KNOWLEDGE SPACES

Project Report Submitted to

Mahatma Gandhi University

In partial fulfilment of the requirement for the award of the degree

MASTER OF SCIENCE IN MATHEMATICS

Submitted By

ELIZABATH JOSEPH P

Reg No. 200011014778

Under The Supervision of

Dr. Seethu Varghese

DEPARTMENT OF MATHEMATICS

Bharata Mata College, Thrikkakara

2020-2022

ACKNOWLEDGEMENT

With great pleasure I express my sincere gratitude to each and every one who helped me for the successful completion of this project. I would like to acknowledge and give my warmest thanks to Dr. Seethu Varghese, Department of Mathematics, Bharata Mata College, Thrikkakara. Her guidance and advice carried me through all the stages of writing my project.

I heartily thank all other teachers of our department for their constant support and guidance. I would also like to give special thanks to my parents and friends for their continuous support and understanding when undertaking my project. Above all, I thank God Almighty for letting me through all the difficulties and guiding me to successfully complete my project.

. **ELIZABATH JOSEPH P**

Place: Thrikkakara

Date: 22-09-2022

DEPARTMENT OF MATHEMATICS

BHARATA MATA COLLEGE

THRIKKAKARA, KOCHI-682021

CERTIFICATE

This is to certify that the project on " KNOWLEDGE SPACES " is a bonafide work done by Ms. ELIZABATH JOSEPH P carried out here in the Department of Mathematics, Bharata Mata College, Thrikkakara, under my supervision, in partial fulfilment of the requirements of the degree of Master of Science in Mathematics of Bharata Mata College, Thrikkakara.

Dr. Seethu Varghese
(Head of Department)

DECLARATION

I hereby declare that the project work entitled "KNOWLEDGE SPACES" submitted to Mahatma Gandhi University, is a record of an original work done by me under the guidance of Dr. Seethu Varghese, Department of Mathematics, Bharata Mata College and this project work is submitted in the partial fulfilment of the requirements for the award of the degree of Master of Science in Mathematics. The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

ELIZABATH JOSEPH P

Place : Thrikkakara
Date : 22-09-2022

CONTENTS

CHAPTER - 4 : Learning Sequences

CHAPTER – 5 : Projection , Decomposition and Adaption of Learning Sequences

ABSTRACT

Knowledge Space Theory relies on a combinatorial viewpoint on the assessment of knowledge.

Chapter 1 begins with the axioms constraining the family K of knowledge states and the further sections give a set-theoretical presentation of the concepts and some results related to it.

Chapter 2 introduces the key concepts of the skill-based knowledge space theory and later discusses its central notions and results. This chapter also emphasises the conceptual distinction between competence states and subsets of skills sufficient for solving an item.

Chapter 3 discusses the competence-based approach which assigns skills and competencies to learning objects for generating the knowledge structures and defines the distributed skill functions and the meshing of knowledge structures.

Chapter 4 shows how to define knowledge spaces by their learning sequences and how to efficiently generate all states of a knowledge space defined from a set of learning sequences. Also, this chapter details the partial order based representation and learning sequence based representation and the efficient algorithms based on it.

Chapter 5 shows how to find a set of learning sequences to represent any given knowledge space and how to use the learning sequence representation as the basis for methods for projecting, adapting and decomposing knowledge spaces.

INTRODUCTION

Knowledge Space Theory (KST) originated with a paper by Doignon and Falmagne (1985). This work was motivated by the shortcomings of the psychometric approach to the assessment of competence. KST is complex and draws from several mathematical disciplines such as combinatorics, statistics and stochastic processes. According to this theory, a student's competence in the subject at a given instant may be identified with students' knowledge state which is a collection of the concepts that the student has mastered. Here, 'concept' means a type of problem that the student has learned to master. For example, in Algebra, consider the problem of solving a quadratic equation with integer coefficients. There are millions of feasible knowledge states for this problem.

The knowledge state of a student reveals exactly what he/she is ready to learn. Each knowledge state has an outer fringe which is the set of concepts that the student in that state may start learning. The student can choose to study a concept in the outer fringe of his/her state. Mastering this concept brings the student to another state. If this is not the maximal state, then it has its own outer fringe and the process continues. The items or problem types form a larger set which is the 'domain' of the body of knowledge. These knowledge states form a particular collection of subsets called the 'knowledge space'.

CHAPTER-1

Basic Concepts

Knowledge structure is a combinatorial structure consists in a family K of subsets of a basic set Q. The elements of Q are the types of problems a student must master while learning the curriculum. The set Q is called the 'domain' of the structure and the elements of Q are referred to as 'items' or 'questions.' The family K is called the 'knowledge space'. The elements of K are called knowledge states.

Every knowledge state in K is a set of items that a student in that state has mastered. A student in state K cannot solve any item in $Q \setminus K$ and moreover the student can solve any item in K. The family K contains all feasible knowledge states in the population of students considered. We suppose that the collection contains at least two subsets: the empty set $\varnothing$ which is that of a student knowing nothing at all in the subject and the full set Q that is the student may have mastered all the items in Q; accordingly, $Q = \cup K$.

1.1 Axioms for Knowledge Spaces

1.1.1 Definition.

A partial knowledge structure is a pair (Q, K) in which Q is the domain of the knowledge structure and K is the knowledge space containing at least Q.
A partial knowledge structure is a knowledge structure if K contains the empty set $\varnothing$.
The knowledge structure (Q, K) is finite when its domain Q is a finite set.
The knowledge structure (Q, K) is discriminative if
$\forall$ items q, p $\in$ Q & $\forall K \in$ K, (q $\in K \Longleftrightarrow$ p $\in K$) $\Rightarrow$q = p

Illustration

Let 10 items be labelled as a, b, …, j. These 10 items are presented in the following table

Table 1.1

Item	**One exemplary instance**
a. Quotients of expressions involving exponents	Simplify the following: a^4b^5
b. Multiplication of two binomials	Write the product below in simplest form: $(t-7)\ (t-4)$
c. Plotting a point in the coordinate plane using a pencil	Using the pencil, plot the point (-8,7).

d. Writing the equation of a line given the slope and a point on the line	A line passes through the point $(x, y) = (-3,5)$ and has a slope of 3. Write an equation for this line.
e. Solving a word problem using a system of linear equations	A scientist has two solutions, which she has labelled Solution A and Solution B. Both A and B contain salt. She knows that A is 40% salt and B is 75% salt. She wants to obtain 60 millilitres of a mixture that is 50% salt. How many millilitres of each solution should she use?
f. Graphing a line when its equation is given	Graph the line $7x + 3y = 24$.
g. Multiplication of a decimal by a whole number	Multiply. 2.95×5
h. Integer addition	$-8 + 4 =$ __ $-7 + 6 =$ __ $-4 + (-4) =$ __
i. Equivalent fractions	Fill in the blank to make the two fractions equivalent: $\frac{3}{5} = \square_{15}$
j. Graphing integer functions	The function h is defined by the rule $h(x) = -x + 2$. Graph h for $x = -1, 0, 1,$ and 2.

The knowledge structure organizing these items is shown in Figure 1.1.

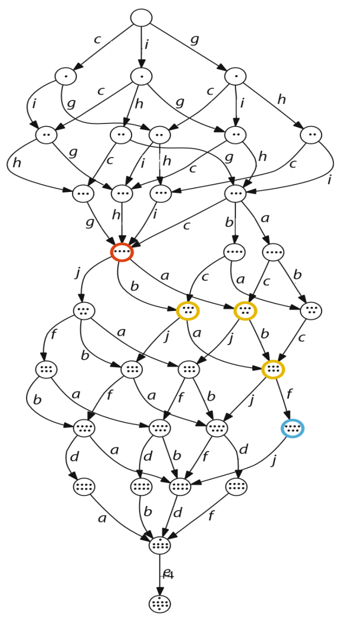

Figure 1.1

The above 10 items generate 34 possible knowledge states which are represented by the circles. The number of dots in each circle denotes the number of items contained in that state. The top circle is the empty state, and learning proceeds from top to bottom, the items being mastered successively. The state represented by the red circle contains the items c, g, h, and i, which can be verified by moving down from the empty circle at the top and following the arrows. According to this graph, there are 16 ways to learn these four items and reach the red state: either item i or item g has to precede item h.

1.1.2 Axioms

A knowledge structure (Q, K) is called a knowledge space if it satisfies the following two conditions:

(L1) LEARNING SMOOTHNESS.

For any two states *K, L* with $K \subset L$, there exists a finite chain of states

$$K_0 = K \subset K_1 \subset \cdots \subset K_n = L.$$

with $K_i = K_{i-1} + \{q_i\}$ and $q_i \in Q$ for $1 \leq i \leq n$.

We have thus $|L \backslash K| = n$.

Example: The state represented by the red circle in Figure 1.1 contains the four items c, i, h, and g while the blue circle contains the same items c, i, h, g, b, a and f. There are intermediate states linking these two states. A student can progress from the red state to the blue state by learning successively b, a and f, passing through some of the yellow states.

(L2) LEARNING CONSISTENCY.

For any two states *K, L* with $K \subset L$, if q not belongs to *K* and $K \cup \{q\} \in$ K, then $L \cup \{q\} \in$ K.

For example, Figure 1.1 shows that a student in the red circle state and blue circle state can learn item j.

By Axiom (L1), all knowledge spaces are finite. Axiom (L2) formalizes the idea that if some item q is learnable by a student in some state *K* which is included in some state *L*, then either q is in *L* or it is learnable by a student in state *L*.

1.1.3 Example.

An example of a knowledge structure H on the domain {a, b, c, d} is given by the equation

H = {∅, {a}, {c}, {a, b}, {a, c}, {c, d}, {a, b, c}, {a, b, d}, {a, c, d}, {a, b, c, d}}

H can be represented by the Hasse diagram of its inclusion relation as follows.

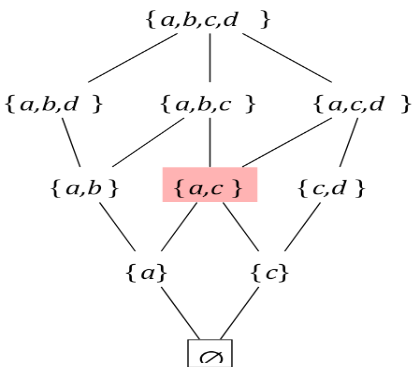

Figure 1.2

The family H is closed under union by axiom(L1) and axiom(L2).

1.1.4 Definition.

Let (Q, K) be a knowledge structure. Then, K is a knowledge space if it is closed under union or ∪-closed, that is, $\cup A \in$ K whenever $A \subseteq$ K.

The dual of a knowledge structure K on Q is the knowledge structure containing all the complements of the states of K.

The symmetric difference between two sets K and L is defined by $K\Delta L = (K\backslash L) \cup (L\backslash K)$

A knowledge structure K is well-graded if, for every two distinct sets *K, L* in K, there exists a natural number h such that $|K\Delta L| = h$ and a finite sequence $K_0 = K, K_1, \ldots, K_n = L$ of sets in K such that $|K_{i-1}\Delta K_i| = 1$ for $1 \leq i \leq h$. The sequence of sets (K_i) is called tight path from *K to L*.

A well-graded knowledge structure is discriminative and is finite.

A family K of subsets of a finite set $Q = \cup K$ is an antimatroid if it satisfies following conditions.

(a) K is closed under union

(b)If *K* is a nonempty subset of the family K, then there is some $q \in K$ such that $K \setminus \{q\} \in$ K

Then, we say the pair (Q, K) is an antimatroid. An antimatroid (Q, K) is finite if Q is finite Also, (Q, K) is a discriminative knowledge structure.

1.1.5 Theorem

For every knowledge structure (Q, K), the following three statements are equivalent.

(i) (Q, K) is a knowledge space.
(ii) (Q, K) is antimatroid
(iii) (Q, K) is a well-graded knowledge space.

1.2 The Base and the atoms

1.2.1 Definition.

The span of a family of sets G is the family containing all sets that are the union of some subfamily of G. Then, G is a $\cup$-closed family.

A base of a $\cup$-closed family K is a minimal subfamily B of K spanning K.

Example: The base of the learning space H displayed in figure1.2 is {{a}, {c}, {a, b}, {c, d}, {a, b, d}}.

Some knowledge spaces have no base. For example, the collection of all the open subsets of the real line.

The empty set does not belong to a base. It is also clear that an element *K* of some base B of K cannot be the union of other elements of B.

1.2.2 Theorem.

Let B be a base for a knowledge space (Q, K). Then $B \subseteq F$ for every subfamily F of states spanning K. Consequently, a knowledge space admits at most one base. Every finite knowledge space has a base.

1.2.3 Definition.

Let K be a knowledge space. For any item *q,* an atom at *q* is a minimal state of K containing *q*. A state *K* is called an atom if it is an atom at *q* for some item q.

1.2.4 Theorem.

Suppose that a knowledge space has a base. Then this base is exactly the collection of all the atoms.

This property plays an essential role in manufacturing a state from any set of items by forming the union of some atoms of these items.

1.3 The Fringe Theorem

1.3.1 Definition.

Let (Q, K) be a knowledge structure. The inner fringe of a state K in (Q, K) is the set of items

$$K^{I} = \{q \in K \mid K \setminus \{q\} \in K\}.$$

The outer fringe of a state K is the set of items

$$K^{O} = \{q \in Q \setminus K \mid K \cup \{q\} \in K\}.$$

Example: (a) The inner fringe and the outer fringe of the blue state of Figure 1.1 is the set containing just item f.

(b)The inner fringe and the outer fringe of the state {a, c} in the knowledge space H, which is shaded red in Figure 1.2 are $\{a, c\}^{I} = \{a, c\}$ and $\{a, c\}^{O} = \{b, d\}$.

1.3.2 Definition.

The fringe of K is the union of the inner fringe and the outer fringe. That is, $K^{F} = K^{I} \cup K^{O}$.

The set of all states whose distance from K is at most n is given by

$$N(K, n) = \{L \in K \mid d(K, L) \leq n\}.$$

N(K, n) is referred to as the n-neighbourhood of the state K.

1.3.3 Theorem.

In a knowledge space, every state is defined by the pair formed of its inner fringe and its outer fringe; that is, there is only one state having these fringes.

Result

A finite knowledge structure is well-graded if and only if every state is defined by its two fringes.

The inner fringe contains the items representing the 'high points' of the student's competence in the topic while the outer fringe contains items that the student is ready to learn.

1.4 The Projection Theorem

Suppose that (Q, K) is a partial knowledge structure. Thus, ∅ is not necessarily in K. Let Q' *be* any proper non empty subset of Q. Define a relation ~ on K by

$$\begin{aligned} K \sim L \quad &\Longleftrightarrow \quad K \cap Q' = L \cap Q' \\ &\Longleftrightarrow \quad K \triangle L \subseteq Q \setminus Q'. \end{aligned}$$

Then, ~ is an equivalence relation on K.

Let [K] denote the equivalence class of ~ containing K, and by $K_{\sim} = \{[K] \mid K \in K\}$ the partition of K induced by ~.

Let (Q, K) be a knowledge structure and take any nonempty Q' in Q. Then, the family $K_{|Q'} = \{W \subseteq Q \mid W = K \cap Q' \text{ for some } K \in K\}$ is the projection of K on Q'. Thus, $K_{|Q'} \subseteq 2^{Q'}$ and the sets in $K_{|Q'}$ may not be states of K.

Example: The items in Figure 1.1 were *a, b, ..., j*. Suppose that we select the subset containing the items *c, d, g*, and *j*. We call those the chosen items and the remaining items the remnants. This operation creates two new kinds of knowledge spaces. The chosen items divide the 34 knowledge states into classes.

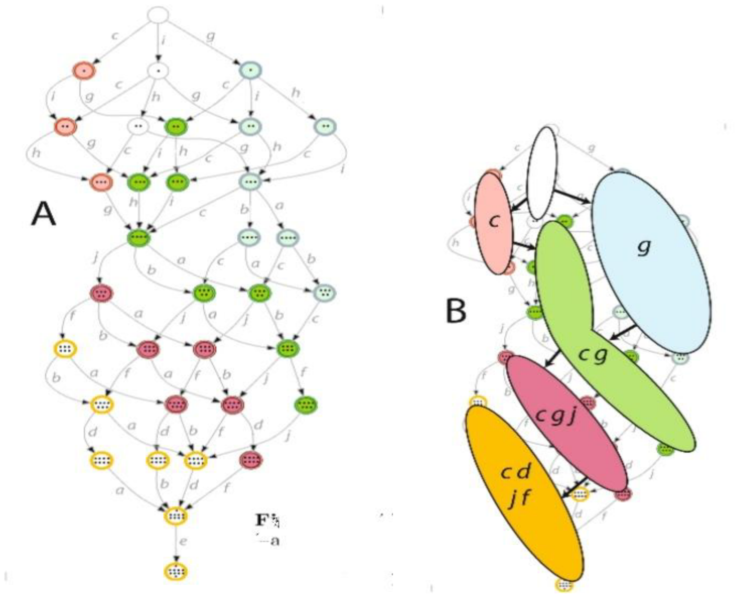

Figure 1.3

The two graphs of Figure1.3 illustrate this operation. Graph A on the left is identical to that of Figure 1.1, except for the colours which indicate the classes of states. Graph B on the right indicates the new knowledge space created by the chosen items. Each of the five coloured regions represents different knowledge states and white ellipse depicts the empty state.

From the empty state at the top, the student can progress by learning first *c* and then *g*, or the other way around. The knowledge states within each of the six classes is also a knowledge space. For example, the state containing *c* and *g* was generated by the six states pictured by the green circles of Graph A, that is,

cig, *cihg,* *cihgb,* *cihga,* *cihgab,* and. *cihgabf.*

Clearly, these six sets of items have *c*, *i* and *g* in common.

Removing these common items, we get a knowledge space

empty state, *h,* *hb,* *ha,* *hab* and *habf*

The knowledge space induced by the knowledge state containing *c* and *g* in the projection knowledge space is given in Figure 1.4.

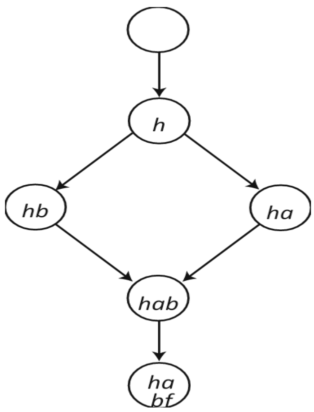

Figure 1.4

For any state K with $\varnothing \not\sim K \in$ K, we define the family

$K_{[K]} = \{L \setminus \cap[K] \mid L \sim K\}$

If $\varnothing \in$ K, we have $K[\varnothing] = [\varnothing]$.

Thus, $K_{[K]} = K_{[L]}$ even if $K \not\sim L$.

1.4.2 Theorem.

Let (Q, K) be a knowledge space with $|Q| \geq 2$. The following two properties hold for every proper nonempty subset Q' of Q :

(i) The projection $K_{|Q'}$ of K on Q' is a knowledge space.

(ii) The children of K are well-graded and $\cup$-closed families.

CHAPTER- 2

Skills, Competencies and Knowledge Structures

Let Q be a knowledge domain that is non empty. Let S be a nonempty set of skills that are relevant for solving the items in Q. Identifying the skills that are sufficient for solving each of the considered items and the set of items which can be solved within a given set of skills is uniquely defined. This subset of items is the possible knowledge state and the corresponding collection forms a knowledge structure.

2.1 The Skill function

2.1.1 Definition

A skill multimap is a triple (Q, S, μ), where μ is a mapping from Q to $2^{2^{\wedge}S}$ and each $\mu(q)$, $q \in$ Q, is a nonempty subsets of S. The elements $C \in \mu(q)$ are called competencies. If the competencies in each $\mu(q)$ are pairwise incomparable, then (Q, S, μ) is called a skill function.

Example:

Let Q = $\{a, b, c, d\}$ and S = $\{s, t, u\}$. Let the skill multimap μ be defined by

$\mu(a) = \{\{s, t\}, \{s, u\}\}$
$\mu(b) = \{\{u\}, \{s, u\}\}$,
$\mu(c) = \{\{s\}, \{t\}\}$
$\mu(d) = \{\{t\}\}$

By definition, either of the skills s and t, or s and u are sufficient for solving item a. The skill multimap μ does not satisfy the incomparability condition as the two competencies in $\mu(b)$ are nested. Thus, (Q, S, μ) is a skill function.

2.1.2 Problem functions

A problem function is a triple (Q, S, p), where p is a mapping from 2^S to 2^Q defined by $p(T) = \{q \in Q \mid \text{there is a } C \in \mu(q) \text{ such that } C \subseteq T\}$ for all $T \subseteq$ S that is monotonic with respect to set inclusion, and satisfies $p(\varnothing) = \varnothing$ and p(S) = Q.

Example:

Let Q = $\{a, b, c, d\}$ and S = $\{s, t, u\}$. Let the skill multimap μ be defined by

$\mu(a) = \{\{s, t\}, \{s, u\}\}$, $\mu(b) = \{\{u\}, \{s, u\}\}$, $\mu(c) = \{\{s\}, \{t\}\}$, $\mu(d) = \{\{t\}\}$

Then p is given by

$p(\varnothing) = \varnothing$ $\quad$ $p(\{s, t\}) = \{a, c, d\}$

$p(\{s\}) = \{c\}$ $\quad$ $p(\{s, u\}) = \{a, b, c\}$

$p(\{t\}) = \{c, d\}$ $\quad$ $p(\{t, u\}) = \{b, c, d\}$

$p(\{u\}) = \{b\}$ $\quad$ $p(S) = Q$

2.1.3 Properties

Given a knowledge domain Q and a skill set S. Let $\mu : Q \to 2^{2^{\wedge}S}$ be a skill function and p its induced problem function. Then the following two statements are equivalent.

(1) For all $q \in$ Q, the competencies $C \in \mu(q)$ are singletons.
(2) $p(T_1 \cup T_2) = p(T_1) \cup p(T_2)$, for all $T_1, T_2 \subseteq$ S.

Generally, a problem function p does not preserve union. But in this case, the combination of two subsets of skills may enable an individual to solve additional items which cannot be solved if only one of the subsets is given. Hence, the inclusion $p(T_1) \cup p(T_2) \subseteq p(T_1 \cup T_2)$ holds for all $T_1, T_2 \subseteq$ S.

Also, the following two statements are equivalent.

(3) For all $q \in$ Q, $\mu(q) = \{C\}$ for some $C \subseteq$ S.
(4) $p(T_1 \cap T_2) = p(T_1) \cap p(T_2)$, for all $T_1, T_2 \subseteq$ S.

(3) implies that each $\mu(q)$ for $q \in$ Q contains only a single competency. The problem function p preserves intersections i.e., $p(T_1 \cap T_2) \subseteq p(T_1) \cap p(T_2)$ for all $T_1, T_2 \in$ S

2.1.4 Definition

Let (Q, S, μ) be a skill function. Then μ is said to be a disjunctive skill function whenever it satisfies (1) and is said to be a conjunctive skill function whenever it satisfies (3).

2.1.5 Delineated Knowledge Structure

Let (Q, S, μ) be a skill multimap and its induced problem function be p. The properties of the problem function p imply that $\varnothing$, Q $\in$ K. Thus, the range of the problem function p forms a knowledge structure (Q, K). This knowledge structure (Q, K) is said to be delineated by the skill multimap μ.

Example*:*

Let Q = $\{a, b, c, d\}$ and S = $\{s, t, u\}$. Let the skill multimap μ be defined by
$\mu(a) = \{\{s, t\}, \{s, u\}\}, \mu(b) = \{\{u\}, \{s, u\}\}, \mu(c) = \{\{s\}, \{t\}\}, \mu(d) = \{\{t\}\}$

Then p is given by $p(\varnothing) = \varnothing, p(\{s\}) = \{c\}, p(\{t\}) = \{c, d\}, p(\{u\}) = \{b\}, p(\{s, t\}) = \{a, c, d\}$, $p(\{s, u\}) = \{a, b, c\}$, $p(\{t, u\}) = \{b, c, d\}$ and p(S) = Q.

By definition, the knowledge structure K on the domain Q delineated by μ is the range of its induced problem function p. It is given by

K = $\{\varnothing, \{b\}, \{c\}, \{c, d\}, \{a, b, c\}, \{a, c, d\}, \{b, c, d\}$, Q$\}$.

2.1.6 Definition

The skill multimaps (Q, S, μ) and (Q, S', μ') on the same knowledge domain Q are said to be isomorphic iff there exists a bijective mapping f from S to S' such that

for all $q \in$ Q, $\mu'(q) = \{f(C) \mid C \in \mu(q)\}$.

2.1.7 Proposition

Two isomorphic skill multimaps (Q, S, μ) and (Q, S', μ') delineate the same knowledge structure (Q, K).

2.1.8 Definition

The skill multimap (Q', S', μ') is said to prolong the skill multimap (Q, S, μ) if the following conditions hold:

1. Q' = Q
2. $S' \supseteq S$
3. $\mu(q) = \{C' \cap S \mid C' \in \mu'(q)\}$.

The skill multimap (Q, S, μ) is minimal if there is no skill multimap delineating the same knowledge structure.

2.1.9 Results

(a) A knowledge space is delineated by a minimal disjunctive skill function if and only if it admits a base.
(b) The two minimal disjunctive skill functions delineating the same knowledge space are isomorphic.

2.2 Korossy's Competence-Performance Approach

Korossy suggested a skill-based extension of knowledge space theory which he calls competence-performance approach. Korossy distinguishes two different levels.

(1) Performance level

It refers to observable behaviour. This level is characterized by a finite knowledge domain Q and a knowledge structure K on Q.

(2) Competence level

It refers to theoretical entities capturing cognitive abilities, the presence or absence of which may explain the observable behaviour. This level is characterized by a finite set S of skills and a collection C of subsets of S. Assume that a competence structure C is a collection of subsets of S containing $\varnothing$ and S. The elements C of C are called competence states.

The competence and performance levels are interconnected by means of two mappings:

(a) Interpretation function *(k)*

It is a mapping k from Q into 2^C which assigns each item a nonempty collection of competence states within which the item can be solved.

(b) Representation function *(r)*

It is a mapping r from C to 2^Q defined by

$$r(C) = \{q \in Q \mid C \in k(q)\}$$

Consider the skill function μ_k from Q into 2^C defined as the reduction $\bar{k}$.

Define a mapping p_k from C to 2^Q as

$p_k(C) = \{q \in Q \mid \text{there is a } C' \in \mu_k(q) \text{ such that } C' \subseteq C\}$.

Clearly, p_k coincides with r. Also, p_k is monotonic with respect to set inclusion and satisfies $p_k(\varnothing) = \varnothing$ and $p_k(S) = Q$. This gives the definition of a problem function with its domain restricted to C. Thus, p_k is the problem function induced by μ_k and the range of p_k is a knowledge structure K on Q, which is called the knowledge structure delineated by μ_k.

2.2.1 Lemma

Let the competence structure C be closed under union. Let $\mu_k : Q \rightarrow 2^C$ be a skill function and $p_k : C \rightarrow 2^Q$ its induced problem function. Then the following two statements are equivalent.

1. For all $q \in Q$, each of the competencies $C \in \mu_k(q)$ is an atom of C.
2. $p_k(C_1 \cup C_2) = p_k(C_1) \cup p_k(C_2)$, for all $C_1, C_2 \in C$.

Proof

Assume 1
Since S is finite, the competence structure C has a base which is formed by the collection of all the atoms.
We know that p_k is a mapping from C to 2^Q defined by $p_k(C) = \{q \in Q \mid \text{there is a } C' \in \mu_k(q) \text{ such that } C' \subseteq C\}$.

Let $q \in p_k(C_1 \cup C_2)$.
Then there is an atom C of C with $C \in \mu_k(q)$ such that $C \subseteq C_1 \cup C_2$ (by our assumption).
Since $C \subseteq C_1 \cup C_2$, $C \subseteq C_1$ or $C \subseteq C_2$
Thus, there is an atom C of C with $C \in \mu_k(q)$ such that $C \subseteq C_1$ or there is an atom C of C with $C \in \mu_k(q)$ such that $C \subseteq C_2$ (by assumption)
This implies $q \in p_k(C_1) \cup p_k(C_2)$
Therefore, $p_k(C_1 \cup C_2) \subseteq p_k(C_1) \cup p_k(C_2)$
By monotonicity of p_k , $p_k(C_1) \cup p_k(C_2) \subseteq p_k(C_1 \cup C_2)$.
Hence, $p_k(C_1 \cup C_2) = p_k(C_1) \cup p_k(C_2)$, for all $C_1, C_2 \in C$.

Conversely, assume 2
We prove this by contradiction.
Suppose C is not an atom of C for some $C \in \mu_k(q)$ and $q \in Q$.
Then we have $C = C_1 \cup C_2$ where $C_1, C_2 \in C$ distinct from C.
By our assumption, $p_k(C_1 \cup C_2) = p_k(C_1) \cup p_k(C_2)$, for all $C_1, C_2 \in C$.
Then, $q \in p_k(C_1 \cup C_2)$ implies $q \in p_k(C_1) \cup p_k(C_2)$.
By definition of $p_k(C_1 \cup C_2)$, there is a $C' \in \mu_k(q)$such that $C' \subseteq C_1 \cup C_2$ and by definition of $p_k(C_1)$, there is a $C' \in \mu_k(q)$ such that $C' \subseteq C_1$
Thus, $C' \subseteq C_1 \subseteq C_1 \cup C_2$ i.e., $C' \subseteq C_1 \subseteq C$ which contradicts the incomparability condition. Hence proved.

2.2.2 Lemma

Let the competence structure C be closed under intersection. Let $\mu_k : Q \rightarrow 2^C$ be a skill function and $p_k : C \rightarrow 2^Q$ be its induced problem function. Then the following two statements are equivalent.

1. For all $q \in Q$, we have $\mu_k(q) = \{C\}$, for some $C \in C$;
2. $p(C_1 \cap C_2) = p(C_1) \cap p(C_2)$, for all $C_1, C_2 \in C$.

Proof

Assume 1
We know that p_k is a mapping from C to 2^Q defined by $p_k(C) = \{q \in Q \mid \text{there is a } C' \in \mu_k(q) \text{ such that } C' \subseteq C\}$.

Let $q \in p_k(C_1) \cap p_k(C_2)$
i.e., $q \in p_k(C_1)$ and $q \in p_k(C_2)$
By our assumption, $\mu_k(q) = \{C\}$ *for some* $C \in$ C.
i.e., there is a $C' =$ C such that $C' \subseteq C_1$ and there is a $C' =$ C such that $C' \subseteq C_2$
i.e., C $\subseteq C_1$ and C $\subseteq C_2$
i.e., C $\subseteq C_1 \cap C_2$
Then, $q \in p_k(C_1 \cap C_2)$
Therefore, $p_k(C_1) \cap p_k(C_2) \subseteq p_k(C_1 \cap C_2)$
By monotonicity of p_k , $p_k(C_1 \cap C_2) \subseteq p_k(C_1) \cap p_k(C_2)$.
Hence, $p_k(C_1 \cap C_2) = p_k(C_1) \cap p_k(C_2)$, for all $C_1, C_2 \in$ C.

Conversely, assume 2.
Suppose the contrary.
Let C_1, C_2 be two distinct elements in $\mu_k(q)$ for some $q \in$ Q.
By our assumption, $p_k(C_1 \cap C_2) = p_k(C_1) \cap p_k(C_2)$.
Let $q \in p_k(C_1 \cap C_2)$
Then, there is a $C \in \mu_k(q)$ such that $C \subseteq C_1 \cap C_2$ which contradicts the incomparability condition.
Hence, for all $q \in$ Q, $\mu_k(q) = \{C\}$ for some $C \in$ C.

2.2.3 Proposition

Let C be a competence structure. Let μ_k : Q $\rightarrow 2^{C}$ be a skill function and p_k : C $\rightarrow 2^{Q}$ be its induced problem function. Let K be the knowledge structure delineated by μ_k. Then the following two statements hold.

(1)If C is closed under union and μ_k, p_k satisfies the equivalent conditions of Lemma 2.2.1, then K is a knowledge space.

(2)If C is closed under intersection and μ_k, p_k satisfies the equivalent conditions of Lemma 2.2.2, then K is a closure space.

<u>Proof</u>

Let $K, L \in$ K such that $K = p_k(C_1)$ and $L = p_k(C_2)$
1)If p_k satisfies condition 2. of Lemma 2.2.1 then, $K \cup L = p_k(C_1) \cup p_k(C_2) = p_k(C_1 \cup C_2) \in$ K for some $C_1, C_2 \in$ C. (since C is closed under union, $C_1 \cup C_2 \in$ C)

Thus, K is a knowledge space.
2)If p_k satisfies condition 2. of Lemma 2.2.2 then, $K \cap L = p_k(C_1) \cap p_k(C_2) = p_k(C_1 \cap C_2) \in$ K for some $C_1, C_2 \in$ C.(since C is closed under intersection, $C_1 \cap C_2 \in$ C)

Hence, K is a closure space.

CHAPTER-3

Competence- Based Approach

The competence-based approach assigns skills and competencies to learning objects for generating the structures for personalized assessment and teaching. These assignments are often established by different experts. The resulting assignments may be distributed among several repositories. The question is whether the skills and competencies assigned to the learning objects can be integrated in a consistent way. This situation is formalized by a distributed skill function, which generalizes the notion of a distributed skill map.

3.1 Distributed skill functions

3.1.Definition

Let (Q, K) be a knowledge structure and let A be a nonempty subset of Q and H = $\{H \in 2^A \mid H = A \cap K$ for some $K \in$ K$\}$. Then, (A, H) is the substructure of the parent structure (Q, K). The substructure H is also referred as the trace of K on A denoted by K$|A$.

3.1.2 Definition

Let (Q_i, S_i, μ_i), $i \in I$ be a collection of skill functions. Then, their merge (Q, S, μ) is defined by

1. $\mathcal{Q} = \bigcup_{i \in I} \mathcal{Q}_i$
2. $\mathcal{S} = \bigcup_{i \in I} \mathcal{S}_i$
3. For all q in $\mathcal{Q}$, $\mu(q) = \bigcup_{i \in I} \mu_i^*(q)$,
   with $\mu^*_i(q) = \mu_i(q)$ if $q \in Q_i$, and $\mu^*_i(q) = \varnothing$ otherwise.

The merge of skill functions need not be a skill function.

3.1.3 Definition

Let (Q_i, S_i, μ_i), $i \in I$, be a collection of skill functions. If their merge (Q, S, μ) is a skill function, then it is called a distributed skill function.

A distributed skill function μ induces a problem function $p : 2^S \rightarrow 2^Q$ in the same way as the component skill functions μ_i induce the problem functions $p_i : 2^{S_i} \rightarrow 2^{Q_i}$

Also, μ delineates a knowledge structure K on the knowledge domain Q together with the component knowledge structure K_i on Q_i delineated by the μ_i.

3.1.4 Example

Let (Q_1, S_1, μ_1) and (Q_2, S_2, μ_2) be two skill functions which are defined on $Q_1 = \{a, b, c, d\}$, $S_1 = \{x, y, z\}$ and $Q_2 = \{a, b, e, f\}$, $S_2 = \{w, x, y\}$ by

$\mu_1(a) = \{\{x, y\}, \{x, z\}\}$ $\qquad$ $\mu_2(a) = \{\{w, y\}, \{x, y\}\}$

$\mu_1(b) = \{\{x\}, \{y\}, \{z\}\}$ $\qquad$ $\mu_2(b) = \{\{w\}, \{x\}\}$

$\mu_1(c) = \{\{x\}, \{y\}\}$ $\mu_2(e) = \{\{x\}, \{w, y\}\}$

$\mu_1(d) = \{\{y, z\}\}$ $\mu_1(f) = \{\{y\}, \{w, x\}\}$

They delineate the following knowledge structures.

$K_1 = \{\varnothing, \{b\}, \{b, c\}, \{a, b, c\}, \{b, c, d\}, Q_1\}$

$K_2 = \{\varnothing, \{b\}, \{f\}, \{b, e\}, \{b, e, f\}, Q_2\}$

Applying the above defined construction provides the distributed skill function with $Q = \{a, b, c, d, e, f\}$, $S = \{x, y, w, z\}$ and

$\mu(a) = \{\{x, y\}, \{x, z\}, \{w, y\}\}$ $\mu(b) = \{\{w\}, \{x\}, \{y\}, \{z\}\}$ $\mu(c) = \{\{x\}, \{y\}\}$
$\mu(d) = \{\{y, z\}\}$ $\mu(e) = \{\{x\}, \{w, y\}\}$ $\mu(f) = \{\{y\}, \{w, x\}\}$

The distributed skill function μ delineates the knowledge structure
$K = \{\varnothing, \{b\}, \{b, c, e\}, \{b, c, f\}, \{a, b, c, e\}, \{b, c, d, f\}, \{b, c, e, f\}, \{a, b, e, f\}, Q\}$ which is illustrated in Figure 3.1.

$K_{1.}$ K_2 .

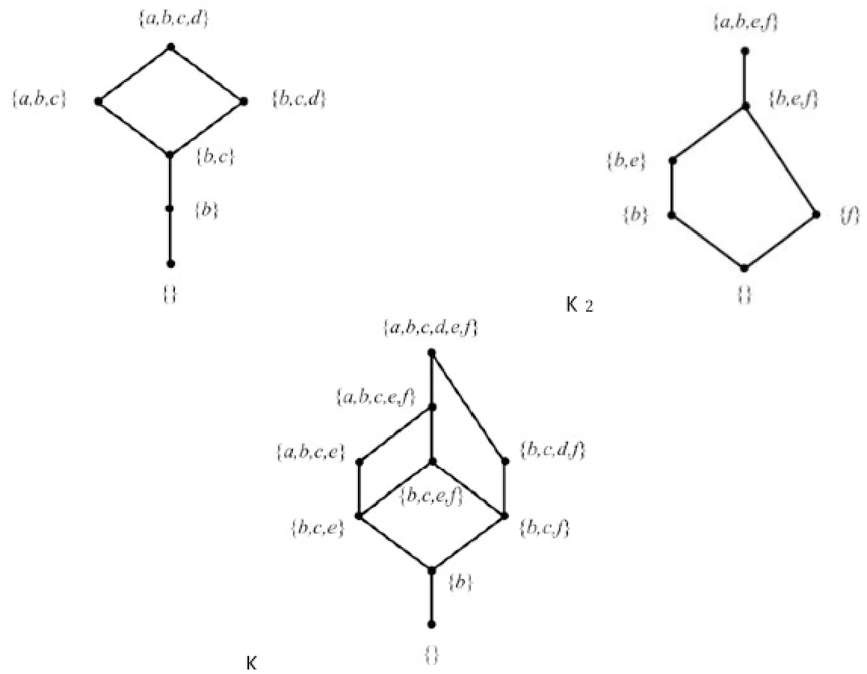

Figure 3.1.

The trace of K on Q_1 and Q_2 yields the substructures

$K|Q_1 = \{\varnothing, \{b\}, \{b, c,\}, \{a, b, c\}, \{b, c, d\}, Q_1\}$

$K|Q_2 = \{\varnothing, \{b\}, \{b, e\}, \{b, f\}, \{a, b, e\}, \{b, e, f\}, Q_2\}$.

Thus, $K_1 = K|Q_1$ and $K_2 \neq \mathcal{K}|_{Q_2}$.

3.2 Meshing Knowledge structures

Falmagne and Doignon introduced the notion of mesh to describe the integration of two knowledge structures on the union of their domains.

3.2.1 Definition

The knowledge structure (Q, K) is said to be a mesh of the knowledge structures (Q_i, K_i), $i \in I$, if

1. $Q = \bigcup_{i \in I} Q_i$;
2. $K_i = K|Q_i$ for all $i \in I$.

Then, the maximal mesh is given by

$$K^{\star} = \{K \in 2^{\bigcup_{i \in I} Q_i} \mid K \cap Q_i \in \mathcal{K}_i, \text{ for all } i \in I\}.$$

When $I = \{1,2\}$, the knowledge structure (Q, K) is a mesh of the knowledge structures (Q_1, K_1) and (Q_2, K_2) if $Q = Q_1 \cup Q_2$, and K_1 and K_2 are the traces of K on Q_1 and Q_2 respectively.

3.2.2 Definition

Let (Q_i, K_i), $i \in I$ be a collection of knowledge structures. Then, it is said to be pairwise compatible if any two of the knowledge structures have the same trace on the intersection of their domains.
i.e. $K_j|Q_j \cap Q_k = K_k|Q_j \cap Q_k$ for all $j, k \in I$.
A collection of knowledge states K_i, $i \in I$, with $K_i \in K_i$ is said to be pairwise compatible if for all $j, k \in I$, $K_j \cap Q_k = K_k \cap Q_j$.

3.2.3 Proposition

Let (Q_i, K_i), $i \in I$ be a collection of knowledge structures. Then, the following statements are equivalent.

1. The knowledge structures are meshable.
2. For all $j \in I$ and all $K_j \in K_j$, there exists a collection (K_i', $i \in I$) of pairwise compatible knowledge states $K_i' \in K_i$ such that $K_j' = K_j$.

3.2.4 Lemma

Let (Q_i, S_i, μ_i), $i \in I$ be a collection of skill functions whose merge is a distributed skill function (Q, S, μ). Let p_i and p be the problem functions induced by μ_i and μ respectively. Then, for all $i \in I$ the following inclusions hold.

1. $\mu_i(q) \subseteq \mu(q) \cap 2^{S_i} \subseteq \mu(q)$ for all $q \in Q_i$;
2. $p_i(T \cap S_i) \subseteq p(T \cap S_i) \cap Q_i \subseteq p(T) \cap Q_i$ for all subsets $T \subseteq S$.
3. $p(T) = \cup p_i(T \cap S_i)$ for all subsets $T \subseteq S$.

3.2.5 Proposition

Let (Q_i, S_i, μ_i), $i \in I$ be a collection of skill functions whose merge is a distributed skill function (Q, S, μ). Let p_i and p be the problem functions induced by μ_i and μ respectively. And the delineated knowledge structures are denoted by K_i and K respectively. Then, for all $i \in I$, the following statements are equivalent.

1. $\mu_i(q) = \mu(q)$ for all $q \in Q_i$;
2. $p_i(T \cap S_i) = p(T) \cap Q_i$ for all subsets $T \subseteq S$.

1&2 implies $K_i = K|Q_i$.

3.2.6 Definition

A collection of skill functions (Q_i, S_i, μ_i), $i \in I$, is consistent whenever the collection of their delineated knowledge structures (K_i, $i \in I$) is meshable.

3.2.7 Definition

Let (Q_i, S_i, μ_i) be a collection of skill functions and p_i be the problem function induced by μ_i where $i \in I$. Then, a collection of skill sets T_i, $i \in I$, with $T_i \subseteq S_i$ is pairwise compatible if the following statements are equivalent.

1. For all $j, k \in I$, $p_j(T_j) \cap Q_k = p_k(T_k) \cap Q_j$;
2. For all $j, k \in I$ and all $q \in Q_j \cap Q_k$, there is a $C_j \in \mu_j(q)$ with $C_j \subseteq T_j$ iff there is a $C_k \in \mu_k(q)$ with $C_k \subseteq T_k$.

3.2.8 Corollary

A collection of skill functions (Q_i, S_i, μ_i), $i \in I$ is consistent iff for all $j \in I$ and any skill set $T_j \subseteq S_j$, there exists a collection (T_i', i $\in I$) of pairwise compatible skill sets $T_i' \subseteq S_i$ such that $p_j(T'_j) = p_j(T_j)$.

3.2.9 Corollary

Consider a collection of skill functions (Q_i, S_i, μ_i), $i \in I$ and the corresponding distributed skill function (Q, S, μ).

1. If the knowledge domains Q_i are pairwise disjoint, then the knowledge structure K delineated by μ is a mesh of the component knowledge structures K_i.

   i.e. $K_i = K|Q_i$, for all $i \in I$.
2. If the skill domains S_i are pairwise disjoint, then $K_i \subseteq K|Q_i$ for all $i \in I$.

3.3 Prerequisite map

3.3.1 Definition

Let L be the set of Learning objects (LOs) that constitute the learning environment. The knowledge domain Q consists of problems that test the skills and competencies taught by the LOs in L. Then, a prerequisite map on the knowledge domain Q is defined as a map $g : Q \rightarrow 2^{2^{\wedge L}}$ that associates to each problem in Q a nonempty collection of nonempty subsets of L.

3.3.2 Definition

A network graph with nodes denoting the learning objects (LOs) is a model of a learning environment. The trails through such a graph are known as LO trails. Navigation through the environment is represented by the paths in the graph. Figure 3.2 is an example of such a graph.

3.3.3 Example

The set of LOs for the learning environment given by L = $\{E, O_1, O_2, O_3, O_4, O_5\}$ is illustrated in Figure 3.2.

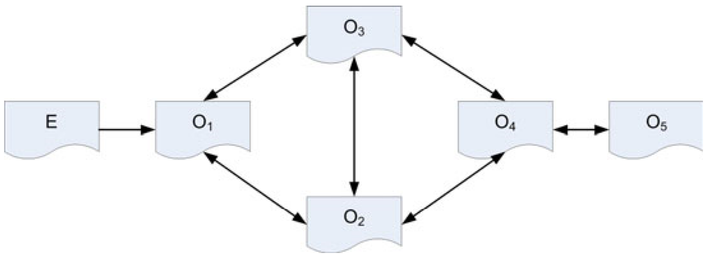

Figure 3.2

Here E denotes an entry point to the learning environment and the LOs O_1 to O_5 teach some content and may be accessed according to the depicted link structure.

Suppose that the problems in Q = $\{a, b, c, d\}$ test the contents taught by the LOs in L. Consider the prerequisite map on Q given by

$$g(a) = \{\{O_1, O_2\}\}$$

$$g(b) = \{\{O_1, O_3\}\}$$

$$g(c) = \{\{O_1, O_2, O_3, O_4\}, \{O_1, O_2, O_5\}\}$$

$$g(d) = \{\{O_1, O_2, O_3, O_5\}\}$$

According to this prerequisite map, the problem *'a'* can be solved using the content taught by the two LOs O_1, O_2. While, problem *'c'* can be solved using the content taught by the LOs O_1, O_2, O_3, O_4, and also by the LOs O_1, O_2, O_5.

3.4 Competence based Approach

Assume that there is a set S of skills that are taught in the learning environment and tested in the knowledge assessment.

The assignment of these skills to LOs in the knowledge domain Q can be formalized by a mapping $\tau : L \rightarrow 2^S$ associating to each LO the set of skills that it teaches. Assume that $\tau(L) = S$.

The assignment of these skills to problems in the knowledge domain Q can be formalized by a mapping μ from Q to 2^{2^S} associating to each problem q in Q a nonempty collection of nonempty subsets of S which is clearly a skill multimap.

Example

Let S = $\{s, t, u, v, w\}$ be the set of relevant skills and L = $\{E, O_1, O_2, O_3, O_4, O_5\}$ be the set of LOs for the learning environment. Suppose that the problems in Q = $\{a, b, c, d\}$ test the contents taught by the LOs in L. Define the mapping τ from L to 2^S by

$\tau(E) = \varnothing$, $\tau(O_1) = \{s\}$, $\tau(O_2) = \{t\}$,

$\tau(O_3) = \{u\}$, $\tau(O_4) = \{v\}$, $\tau(O_5) = \{w\}$.

Let the skill multimap μ be given by

$\mu(a) = \{\{s, t\}\}$, $\mu(b) = \{\{s, u\}\}$,

$\mu(c) = \{\{s, t, u, v\}, \{s, t, w\}\}$, $\mu(d) = \{\{s, t, u, w\}\}$.

Then following the route outlined above we get the prerequisite map on Q as follows

$g(a) = \{\{O_1, O_2\}\}$

$g(b) = \{\{O_1, O_3\}\}$

$g(c) =\{\{O_1, O_2, O_3, O_4\},\{O_1, O_2, O_5\}\}$

$g(d) = \{\{O_1, O_2, O_3, O_5\}\}$

Result

The skill multimap μ delineates a knowledge structure on Q.

Example: The knowledge structure K for the above example is given by

$K = \{\varnothing, \{a\}, \{b\}, \{a, b\}, \{a, c\}, \{a, b, c\}, \{a, b, d\}, Q\}$,

This is illustrated in Figure 3.3.

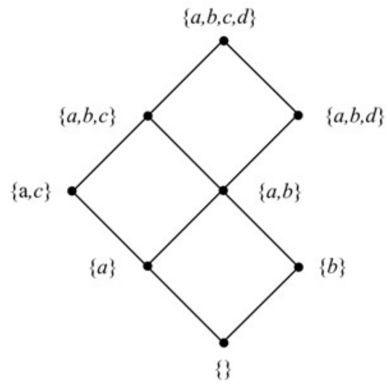

Figure 3.3

The figure 3.3 shows the possible learning paths of the above learning environment.

3.5 Compatibility conditions

3.5.1 Compatibility conditions

Let $L_0, ..., L_n$ be a trail of visited LOs from L and

K_0 and K_n denote the pre-assessment before the LOs are inspected and post-assessment after finishing the interaction with the learning environment. Let g be a prerequisite map on Q that gives the relation between the LOs and assessment items. The conditions C1, C2, and C3 stated below give a minimal set of compatibility requirements.

C1. An item cannot be solved before visiting a set of relevant LOs.

i.e., $q \in K_t \setminus K_0$ implies that there is a subset $M \in g(q)$ such that $M \subseteq \{L_0, ..., L_t\}$.

C2. An item can be solved only after visiting a relevant LO.

i.e., $q \in K_t \setminus K_{t-1}$ implies that there is a subset $M \in g(q)$ such that $L_t \in M$.

C3. There is no forgetting. i.e., the trail of knowledge states $K_0, ..., K_n$ is increasing with respect to set-inclusion.

i.e., $K_0 \subseteq ... \subseteq K_n$

3.5.2 Definition

A trail $(L_0, K_0), ..., (L_n, K_n)$ on L × K is called consistent if the trail of knowledge states $K_0, ..., K_n$ is compatible with the trail $L_0, ..., L_n$ of LOs.

i.e., whenever the compatibility conditions C1-C3 holds.

3.5.3 Definition

The scope $s(L_0, ..., L_n) \in$ K of the trail $L_0, ..., L_n$ is the set of items $q \in$ Q for which there exists $M \in g(q)$ such that $M \subseteq \{L_0, ..., L_n\}$.

3.5.4 Strict Learning Assumption (SLA)

Given a trail $L_0, ..., L_n$ of LOs from L. Let g be a prerequisite map g on Q and $K_0, K_n \subseteq$ Q. Consider a trail $K_0, ..., K_n$ of knowledge states in the domain Q, which is defined as follows :

For all $q \in K_n \setminus K_0$ and $t \in \{1, ..., n-1\}$,
$q \in K_t \setminus K_0$ iff there is a subset $M \in g(q)$ such that $M \subseteq \{L_0,...,L_t\}$.

This condition is called Strict Learning Assumption (SLA).

3.5.5 Weak Learning Assumption (WLA)

Given a trail $L_0, ..., L_n$ of LOs from L. Let g be a prerequisite map g on Q and $K_0, K_n \subseteq$ Q. Consider a trail $K_0, ..., K_n$ of knowledge states in the domain Q, which is defined as follows:
For all $q \in K_n \setminus K_0$ and $t \in \{1, ..., n-1\}$,
$q \in K_t \setminus K_0$ iff there is a subset $M \in g(q)$ such that $M \subseteq \{L_0, ..., L_t\}$ and $M \cap \{L_{t+1}, ..., L_n\} = \varnothing$.

This condition is called Weak Learning Assumption (WLA).

3.5.6 Proposition

Given an observable trail $L_0, ..., L_n$ of LOs from L, a prerequisite map l on Q, and $K_0, K_n \subseteq Q$. Let $K_0, ..., K_n$ be a trail of knowledge states in the domain Q, which is defined in the following way:

For all $q \in K_n \setminus K_0$ and $t \in \{1, ..., n-1\}$, $q \in K_t \setminus K_0$ iff there exists $M \in g(q)$ such that $M \subseteq \{L_0, ..., L_t\}$.

Then the trail $K_0, ..., K_n$ satisfies the compatibility conditions C1-C3.

Proof

From the given condition, C1 holds.

To prove C2
let $q \in K_t \setminus K_{t-1}$, for some $t \in \{1, ..., n\}$

i.e., $q \in K_t$ and $q \notin K_{t-1}$

Then, q does not belong to K_0.

Then, $q \in K_t \setminus K_0$

By the given condition, there exists $M \in g(q)$ such that $M \subseteq \{L_0, ..., L_t\}$.

Thus, for all $M \in g(q)$, $M \not\subseteq \{L_0, \ldots, L_{t-1}\}$

i.e., $M \subseteq \{L_0, ..., L_t\}$ and $M \not\subseteq \{L_0, \ldots, L_{t-1}\}$

i.e., $L_t \in M$

Thus, C2 holds.

To prove C3

Assume that $q \in K_{t-1} \setminus K_0$, for some $t \in \{1, ..., n\}$.

By the given condition, there exists a subset $M \in g(q)$ such that $M \subseteq \{L_0, ..., L_{t-1}\}$.

Then, $M \subseteq \{L_0, ..., L_{t-1}, L_t\}$

By the given condition, $q \in K_t \setminus K_0$

Therefore, $K_{t-1} \subseteq K_t$

Thus, C3 holds.

3.5.7 Proposition

Given an observable trail $L_0, ..., L_n$ of LOs from L, a prerequisite map l on Q, and $K_0, K_n \subseteq Q$. Let $K_0, ..., K_n$ be a trail of knowledge states in the domain Q, which is defined in the following way:

For all $q \in K_n \setminus K_0$ and $t \in \{1, ..., n-1\}$,

$q \in K_t \setminus K_0$ iff there exists $M \in g(q)$ such that $M \subseteq \{L_0, ..., L_t\}$ and $M \cap \{L_{t+1}, ..., L_n\} = \varnothing$.

Then $K_0, ..., K_n$ satisfies the compatibility conditions C1-C3.

<u>Proof</u>

From the given condition, C1 holds.

To prove C2

Let $q \in K_t \setminus K_{t-1}$ for some $t \in \{1,...,n\}$

i.e., $q \in K_t$ and $q \notin K_{t-1}$

Then, q does not belong to K_0.

Then, $q \in K_t \setminus K_0$.

By the given condition, there exists $M \in g(q)$ such that $M \subseteq \{L_0,...,L_t\}$ and $M \cap \{L_{t+1},...,L_n\} = \varnothing$.

Thus, for all M $\in g(q)$, $M \not\subseteq \{L_0, \ldots, L_{t-1}\}$ or $M \cap \{L_t, ..., L_n\} \neq \emptyset$.

i.e., $M \subseteq \{L_0, ..., L_t\}$ and $M \not\subseteq \{L_0, \ldots, L_{t-1}\}$

This implies $L_t \in M$.

Also, $M \cap \{L_{t+1}, ..., L_n\} = \varnothing$ and $M \cap \{L_t, ..., L_n\} \neq \emptyset$

which implies $L_t \in M$.

Thus, C2 holds

To prove C3

Assume that $q \in K_{t-1} \setminus K_0$, for some $t \in \{1,...,n\}$.
By the given condition, there exists a subset $M \in g(q)$ such that $M \subseteq \{L_0, ..., L_{t-1}\}$ and $M \cap \{L_t, ..., L_n\} = \varnothing$.
Then, $M \subseteq \{L_0, ..., L_{t-1}, L_t\}$ and $M \cap \{L_{t+1}, ..., L_n\} = \varnothing$
By the given condition, $q \in K_t \setminus K_0$
Therefore, $K_{t-1} \subseteq K_t$
Thus, C3 holds.

CHAPTER -4

Learning Sequences

Learning sequences are sequences of steps through which a student, starting with no knowledge, could learn all the concepts in the space. A knowledge space may have an infinite number of possible learning sequences, but it is possible to correctly and accurately represent any knowledge space using a subset of these sequences. We will learn in this chapter how to efficiently generate all states of a knowledge space defined from a set of learning sequences.

4.1 Knowledge Spaces

4.1.1 Axioms

Consider the Knowledge space K as a finite family of finite sets where each finite set represents the knowledge state *K* of a student with elements representing the concepts that a particular student has mastered.

A set is said to be a valid knowledge state if it belongs to the family K. Then, K may satisfy some or all of the following axioms:

1.Accessibility or Downgradability

A family K of sets is accessible if every nonempty set $S \in$ K has an element $x \in S$ such that $S \setminus \{x\} \in$ K.

2.Union Closure

A family K of sets is closed under union if for every two sets S and T in K, $S \cup T \in$ K.

3.Intersection Closure

A family K of sets is closed under intersection if for every two sets S and T in K, $S \cap T \in$ K.

Accessibility formalizes the notion that every state of knowledge can be reached by learning one concept at a time. Union closure formalizes the notion that the knowledge of two individuals may be pooled to form a state of knowledge that is also valid.

4.1.2 Definition

A knowledge space is a finite family of finite sets that is accessible and closed under unions.

A quasi-ordinal space is a knowledge space that is closed under intersections.

4.2 Quasi-ordinal spaces from partial orders

4.2.1 Definition

A partial order is a relation < among a set of objects, satisfying irreflexivity ($x \nless x$) and transitivity ($x < y$ and $y < z$ implies $x < z$). The Hasse diagram of a partial order is a directed graph containing an edge $x \rightarrow y$ whenever $x < y$ and there does not exist z with $x < z < y$. In the partial order, $x < y$ iff there exists a directed path from x to y in the Hasse diagram.

4.2.2 Definition

A set S forms a valid state of knowledge in quasi-ordinal spaces iff it is a lower set, that is, for every edge $x \rightarrow y$ of the Hasse diagram, either $x \in S$ or y not in S. Figure 4.1 shows a Hasse diagram on eight concepts and the 19 states in the quasi-ordinal space derived from it.

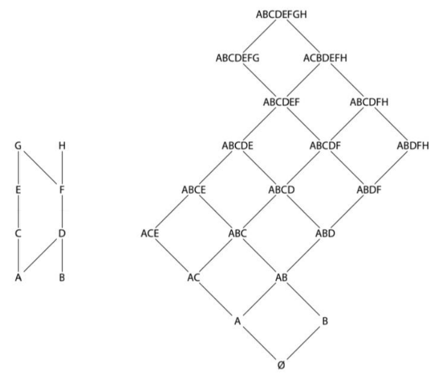

Figure 4.1

4.2.3 Fringes of quasi-ordinal spaces

In a quasi-ordinal space, the inner fringe of S consists of the maximal concepts in S and the outer fringe consists of the minimal concepts not in S.

The fringe of S can be calculated easily from its Hasse diagram. Let F be the domain. Remove x from F whenever there is an edge $x \rightarrow y$ with $y \in S$, and remove y from F whenever there is an edge $x \rightarrow y$ with x not in S. The remaining set at the end of this procedure is the fringe.

4.2.4 Generating the states of a quasi-ordinal space

Assume that the edges of the Hasse diagram are prerequisite relations between concepts.
ie, if x and y are concepts represented as vertices in a Hasse diagram containing the edge $x \rightarrow y$, then y cannot be learned unless x is learnt.

Reverse search method

Reverse search depends on having a predecessor relationship among the states of the given quasi-ordinal space which is defined as follows:
Choose arbitrarily a sequence of the concepts such that, if $x < y$ in the order, then x must appear prior to y in the sequence. For every state S of the quasi-ordinal space derived from the partial order, let x be the concept in S that is latest in the sequence. Then $S \setminus \{x\}$ is also a state, which is the predecessor of S. Continuing this process will lead to the empty set. Then, the graph formed by connecting each state to its predecessor is a rooted tree with root the empty set.

Depth first traversal method

Next, we perform the depth first traversal of this tree as follows:
For each state S of the traversal, we maintain a set children (S) of concepts that may be added to S to form a child of S. The successors of each state are found recursively. To calculate

children($S \cup \{x\}$) from children(S), remove from children(S) any y occurring prior to x in the topological order and add to children(S) any concept y reachable from x by a Hasse diagram edge $x \rightarrow y$ such that all other prerequisites of y already belong to S. Then, output $S \cup \{x\}$ and continue recursively to next state $S \cup \{x, y\}$ for each y in children ($S \cup \{x\}$) and so on.

Prerequisite test

Method 1:
To determine whether all prerequisites of y belong to a set S, count the prerequisites of y that do not belong to the current state of the traversal. Whenever the traversal steps from a state S to next state $S \cup \{x\}$ and x is a prerequisite of y, decrement this count, and whenever the traversal returns from $S \cup \{x\}$ to S, increment this count. If all prerequisites of y belong to the current state S, then the count for y is zero. The time spent to update these counts when we step from a state S to a state $S \cup \{x\}$ or vice versa is proportional to the number of Hasse diagram edges that go out of x.

Method 2 :
A state S is represented as a bitvector with 1 in ith position if the ith concept belongs to S and with 0 in other positions. Whenever the traversal steps from a state S to next state $S \cup \{x\}$ or vice versa, change the bit corresponding to the concept x. To check whether all prerequisites of y belong to the state S, perform a bitwise Boolean or function between two bitvectors where one represents S and the other has a 0 in each prerequisite position and a 1 in other positions. If the resulting bitvector has 1 in all its positions, then all prerequisites of y belong to the current state S. The time required for each prerequisite test using the bitvector method is proportional to the number of concepts in the knowledge space.

4.3 Knowledge Spaces from Learning Sequences

Given a knowledge space. Then, there may be many orderings through which a student, starting with no knowledge, could learn all the concepts in the space. Such an ordering is called a learning sequence.

4.3.1 Definition

A learning sequence is an injective map σ from the integers $\{0,1,2, ...n-1\}$ to the n concepts forming the domain of a knowledge space, with the property that each prefix $P_i(\sigma) = \sigma(\{0,1, 2, ...i-1\})$ is a valid knowledge state in the knowledge space.

4.3.2 Example

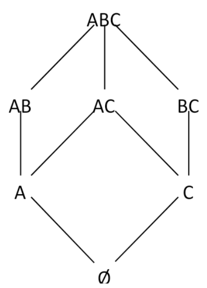

Figure 4.2

The knowledge space depicted in Figure 4.2 has four learning sequences:
(1) *A, B, C* (2) *A,C,B* (3) *C,A,B* (4) *C,B,A*

4.3.3 Results

(a) Learning sequences can be interpreted as shortest paths from the empty set to the set of all concepts in the graph with knowledge states as vertices and the connected pairs of states differ only by a single concept.

Example: In Figure 4.2, the learning sequence *A, C, B* corresponds to the path $\varnothing$, $\{A\}$, $\{A,C\}$, $\{A,B,C\}$ and vice versa.

(b) Let Σ be a set of learning sequences.
Then, all prefixes of Σ are knowledge states in the associated knowledge space. By the union-closure property of knowledge spaces, union of prefixes of Σ must also be a knowledge state in the associated knowledge space.
Example: In example 4.3.2, *A, B, C* and *C, B, A* are two learning sequences. The prefixes of these sequences are the six sets $\varnothing$, $\{A\}$, $\{A, B\}$, $\{A, B, C\}$, $\{C\}$, and $\{B, C\}$. By forming unions of prefixes, we get the seventh set $\{A\} \cup \{C\} = \{A, C\}$.

4.3.4 Definition

Let Σ be a set of learning sequences over a finite set of concepts. We define L_Σ to be the family of sets that are unions of prefixes of sequences in Σ.

4.3.5 Theorem

For any set Σ of sequences, L_Σ is a knowledge space.

<u>Proof</u>

To prove L_Σ is a knowledge space.
So, it is enough to prove that L_Σ is both closed under unions and accessible.
Let two sets $S, T \in L_\Sigma$.
By the definition of L_Σ, S and T are unions of sets of prefixes of sequences in Σ.
From result 4.3.3(b), we get $S \cup T$ is also a union of sets of prefixes of sequences in Σ.
Thus, $S \cup T \in L_\Sigma$.
Hence, L_Σ is closed under unions.
Let a nonempty set $S \in L_\Sigma$.
i.e., S is a union of sets of prefixes of sequences in Σ.
Let $P_i(\sigma_j)$ be a set of prefixes in S whose total length is minimum.
Suppose $P_i{-}1(\sigma_j)$ is a set of prefixes of sequences in Σ such that its union is $T \in L_\Sigma$.
Then, $T \neq S$. Otherwise, S will contain a set of prefixes whose total length is less than that of $P_i(\sigma_j)$ which is a contradiction.
By the definition, $P_i(\sigma_j) = \sigma_j(\{0,1,2,...,i-2, i-1\})$ and $P_i(\sigma_j) = \sigma_j(\{0,1,2,...i-2\})$.
Clearly, T differs from S only by a single element $\sigma_j(i{-}1)$.
i.e., $S \backslash \sigma_j(i{-}1) = T \in L_\Sigma$
Thus, L_Σ is accessible.
Hence, L_Σ is a knowledge space.

4.3.6 Theorem

Let S be a state of a knowledge space K. Then there is a learning sequence of K containing S as one of its prefixes.

Proof

Suppose S is a state of a knowledge space K.
Since K is a knowledge space, K is accessible.
Thus, every nonempty set $S \in$ K has an element $x \in S$ such that $S \setminus \{x\} \in$ K.
On repeatedly applying this, we will get at least one learning sequence of K, say σ.
Since each prefix is a knowledge state in the associated knowledge space, $P_i(\sigma)$ is a knowledge state in K.
Thus, S and $P_i(\sigma)$ are knowledge states of K.
Therefore by union-closure property, $S \cup P_i(\sigma) \in$ K
Then, $P_0(\sigma)$, $P_1(\sigma)$, ..., S, S $\cup$ $P_0(\sigma)$, S $\cup$ $P_1(\sigma)$, ... forms a sequence of sets.
Removal of duplicates from this sequence results in a sequence of sets having S as one of its prefixes.

4.3.7 Definition

An atom of a knowledge space is a nonempty state S for which there exists only one concept $x \in S$ such that $S \setminus \{x\}$ is a valid state.

4.3.8 Lemma

A nonempty state S of a knowledge space K is an atom if and only if it cannot be formed as the union of two smaller states.

Proof

Necessary part

Suppose S is an atom of a knowledge space K.
Assume the contrary.
i.e., S can be formed as the union of two smaller states.
i.e., $S = T \cup U$ where T and U are both smaller than S.
By theorem 4.3.6, there is a learning sequence containing T and S as prefixes.
Since K is a knowledge space, K is accessible.
Also, $T, U \in$ K
Thus, $T \in$ K has an element $x \in T$ such that $T \setminus \{x\} \in$ K and $U \in$ K has an element $y \in U$ such that $U \setminus \{y\} \in$ K.

Therefore, $x \in S$ such that $S \setminus \{x\} \in$ K and $y \in S$ such that $S \setminus \{y\} \in$ K.
i.e., there exists $x, y \in S$ such that $S \setminus \{x\} \in$ K and $S \setminus \{y\} \in$ K.
Then, S is not an atom which is a contradiction.
Hence, S cannot be formed as the union of two smaller states.

Sufficient part

Suppose S cannot be formed as the union of two smaller states.
Assume the contrary.
i.e., S is not an atom.
Then, there exists $x, y \in S$ such that $S \setminus \{x\} \in$ K and $S \setminus \{y\} \in$ K.
i.e., Clearly, $S \setminus \{x\} \cup S \setminus \{y\} = S$ which is a contradiction.
Hence, S is an atom.

4.3.9 Theorem

Let K be any knowledge space and let Σ be a set of learning sequences of K. Then, $L_\Sigma \subseteq$ K with equality iff every atom of K is one of the prefixes of Σ.

Proof

Let K be any knowledge space and let Σ be a set of learning sequences of K.

By the definition of learning sequences, each prefix of a sequence in Σ is a state of K.

Since L_Σ is the family of sets that are unions of prefixes of sequences in Σ and K satisfies union-closure property, $L_\Sigma \subseteq$ K.

Necessary part

We prove this by contrapositive method.
Suppose there is an atom S of K which is not a prefix of Σ.
By lemma 4.3.8, S cannot be written as a union of prefixes of Σ.
Thus, S does not belong to L_Σ.
Hence, L_Σ differs from K.

Sufficient part

Suppose every atom of K is one of the prefixes of Σ.
To prove $L_\Sigma =$ K.
We have $L_\Sigma \subseteq$ K.
So, it is enough to prove that K $\subseteq L_\Sigma$.
Let S be a knowledge state of K.
By theorem 4.3.6, corresponding to each state S of K , there is a learning sequence of K containing S as one of its prefixes.

Then, $S \in L_\Sigma$.
Thus, K $\subseteq L_\Sigma$.
Hence, $L_\Sigma =$ K.

4.4 Representation of knowledge spaces by numerical vectors

4.4.1 Definition

Given a state S in a knowledge space K and a set $\Sigma = \{\sigma_0, \sigma_1, ..., \sigma_{k-1}\}$ of k learning sequences within the space. Then, $mex_i(S)$ is defined as the concept in σ_i with a minimum index that is excluded from S.
i.e., $mex_i(S) = \min\{j \mid \sigma_i(j)$ does not belong to $S\}$. Then, mex(S) is defined as the vector mex(S) = ($mex_0(S)$, $mex_1(S)$, ..., $mex_{k-1}(S)$).

4.4.2 Example

The knowledge space L_Σ generated from three sequences *ABCDEF*, *BDFCAE*, and *CBEFAD* is depicted in Figure 4.3.

Let σ_0 = *ABCDEF,* σ_1 = *BDFCAE* and σ_2 = *CBEFAD.*
Consider the knowledge state *ABDF*.
Then,
$mex_0(ABDF) = \min\{j \mid \sigma_0(j)$ does not belong to $ABDF\}$
= min {2,3} (since *C, E* does not belongs to *ABDF*)
= 2
$mex_1(ABDF) = \min\{j \mid \sigma_1(j)$ does not belong to $ABDF\}$
= min {3,5} (since *C, E* does not belongs to *ABDF*)
= 3
$mex_2(ABDF) = \min\{j \mid \sigma_2(j)$ does not belong to $ABDF\}$
= min {0,2} (since *C, E* does not belongs to *ABDF*)
= 0
Therefore, mex(*ABDF*) = ($mex_0(ABDF)$, $mex_1(ABDF)$, $mex_2(ABDF)$)
= (2, 3, 0)
Similarly, we can find mex(S) for all knowledge states S in L_Σ.
Each state S along with its mex(S) is given in the figure below.

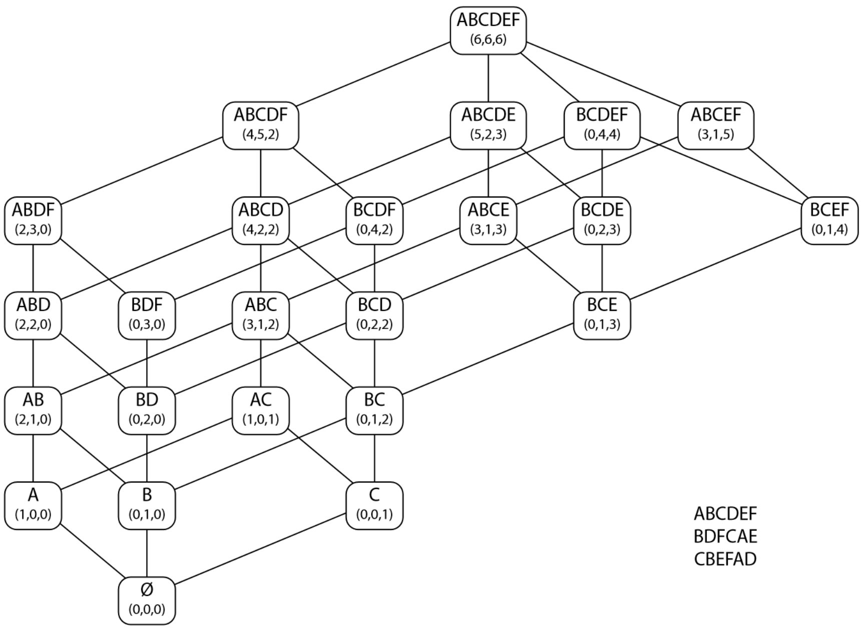

Figure 4.3

4.4.3 Result

If S is the whole domain, then we define $\text{mex}_i(S)$ as $\text{mex}_i(S) = n$. Therefore, $\text{mex}_i(S)$ is the size of the largest prefix of learning sequence σ_i that is a subset of S. Thus, the mex function maps states in the knowledge space to vectors in Z_k.

4.4.4 Definition

The mapping up(v) from vectors in Z_k to states in the knowledge space is defined by up($v) = \bigcup P_{v_i}(\sigma_i)$, $0 \le i < k$.

For any set S, up(mex(S)) $\subset S$.

Also, up(mex(S)) = S iff S is a state in the knowledge space L_Σ.
We can calculate mex($S \setminus \{x\}$) easily if mex(S) is known, by taking the coordinate-wise minimum of mex(S) and the positions of x.

4.5 Generating the states of a knowledge Space

4.5.1 A tree of states

We can list all states of a knowledge space by an algorithm based on exploring a tree with knowledge states as nodes.

The parent of each node is found by a predecessor operation. To generate the predecessor of a state, repeatedly reduce the last non zero coordinate of its mex vector until the up function maps the reduced vector to a set different from the state itself.

In order to find the successors of a state S, we need to perform the following steps:

1. Set p to the smallest value of i such that S is the union of prefixes from the first i sequences.
2. For each $i \geq p$, such that $\sigma_i(\text{mex}_i(S)) \neq \sigma_j(\text{mex}_j(S))$ for all $j < i$ (i.e., the first excluded concept in σ_i differs from the first excluded concept in all earlier sequences):
    a) Let v be formed from mex(S) by adding one to its ith coordinate.
    b) Let T = up(v) = $S \cup \{\sigma_i(\text{mex}_i(S))\}$ and output T.

This algorithm generates the states that have S as their predecessor.

4.5.2 Recursive tree search

To generate all states in the knowledge space, we perform a depth first traversal of the states of the space, starting from the empty set to find the successors of each state. Then, we search recursively for each successor found. If T is a successor of S, then the value of p needed to search the successors of T equals the value of i used when T is generated from S. However, the mex values of S and T may differ not just in the ith coordinate but also in other coordinates greater than i. Continuing this process, we will get all the states of the knowledge space. If the knowledge space is generated by k learning sequences, then the time per state is $O(k)$ except for the time to calculate the mex value of each new state.

CHAPTER- 5

Projection, Decomposition and Adaption of Learning Sequences

In this chapter, we show how to use learning sequences to derive one knowledge space from another:

- We show how to project a knowledge space with a large number of concepts onto a smaller knowledge space that uses a sampled subset of the concepts.
- We show how to decompose a knowledge space into a join of simpler knowledge spaces and how to find the smallest possible set of learning sequences that defines a given knowledge space.
- We show how to adapt a knowledge space by adding or removing individual states that belong to its inner and outer fringe.

5.1 Projection

Firstly, we find a small sample of concepts. For this, we consider the sampling scheme that uses distance measure between pairs of concepts. Then, projects the given knowledge spaces onto smaller knowledge spaces.

We now define the distance between two concepts as the sum of the distances between them in each of the learning sequences defining the given knowledge space.

If F is any family of sets and P is any subset of $\cup$F, then the projection is defined by
$F_P = \{S \cap P \mid S \in F\}$
which is the set of intersections of each state in F with P.

5.1.1 Theorem

If F is a knowledge space, and P is any subset of $\cup$F, then F_P is also a knowledge space.

Proof

To prove union-closure property of F_P.
Let $S_1, S_2 \in F$.
Then, $S_1 \cap P, S_2 \cap P \in F_P$
Consider $(S_1 \cap P) \cup (S_2 \cap P) = (S_1 \cup S_2) \cap P$.
Since is F is a knowledge space, $S_1 \cup S_2 \in F$.
Thus, $(S_1 \cap P) \cup (S_2 \cap P) \in F_P$.
Hence, union- closure property holds for F_P.
To prove the accessibility property of F_P Repeatedly use accessibility in F, to remove concepts from S until a concept, say x in P is removed.

Then, x $\in S \cap P$.

By accessibility property of F, $S\backslash\{x\}$, $P\backslash\{x\} \in$ F
Clearly, $S \cap P\backslash\{x\} \in F_P$.
Hence, accessibility property holds for F_P.
To prove intersection-closure property of F_P.
Let $S_1, S_2 \in$ F.
Then, $S_1 \cap P, S_2 \cap P \in F_P$
Consider $(S_1 \cap P) \cap (S_2 \cap P) = (S_1 \cap S_2) \cap P$.
Since F is a knowledge space, $S_1 \cap S_2 \in$ F.
Thus, $(S_1 \cap P) \cap (S_2 \cap P) \in F_P$.
Hence, intersection-closure property holds for F_P.
Therefore, F_P is a knowledge space.

5.1.2 Theorem

Let F be a knowledge space, defined from a collection of learning sequences σ_i, and let P be a subset of the concepts in F. For each learning sequence σ_i , let σ_i' be the subsequence of σ_i consisting only of the concepts in P, in the same order as they appear within σ_i. Then the set of learning sequences σ_i' define the knowledge space F_P.

Proof

Given that for each learning sequence σ_i , σ_i' is the subsequence of σ_i consisting only of the concepts in P.

Then, $Pj(\sigma_i') = P \cap Pj(\sigma_i)$ where $Pj(\sigma_i) \in$ F.
i.e, $Pj(\sigma_i') \in F_P$
Since F_P is a knowledge space, union of prefixes of sequences σ_i' is also a state of F_P.

Conversely,
Let S be any state of F_P.
Then $S = T \cap P$ where T is a state of K.
Since K is defined from the learning sequences σ_i, T can be formed as a union of prefixes of these learning sequences.
On intersecting each of these prefixes with P gives a representation of S as a union of prefixes of the learning sequences σ_i'.
i.e., every state of F_P is also a state of the knowledge space defined by the sequences σ_i'.
Hence, the set of learning sequences σ_i' define the knowledge space F_P .

5.1.3 Example

Figure 5.1 shows a set of learning sequences defining the original knowledge space K and their restriction to a set of learning sequences defining the projected space.

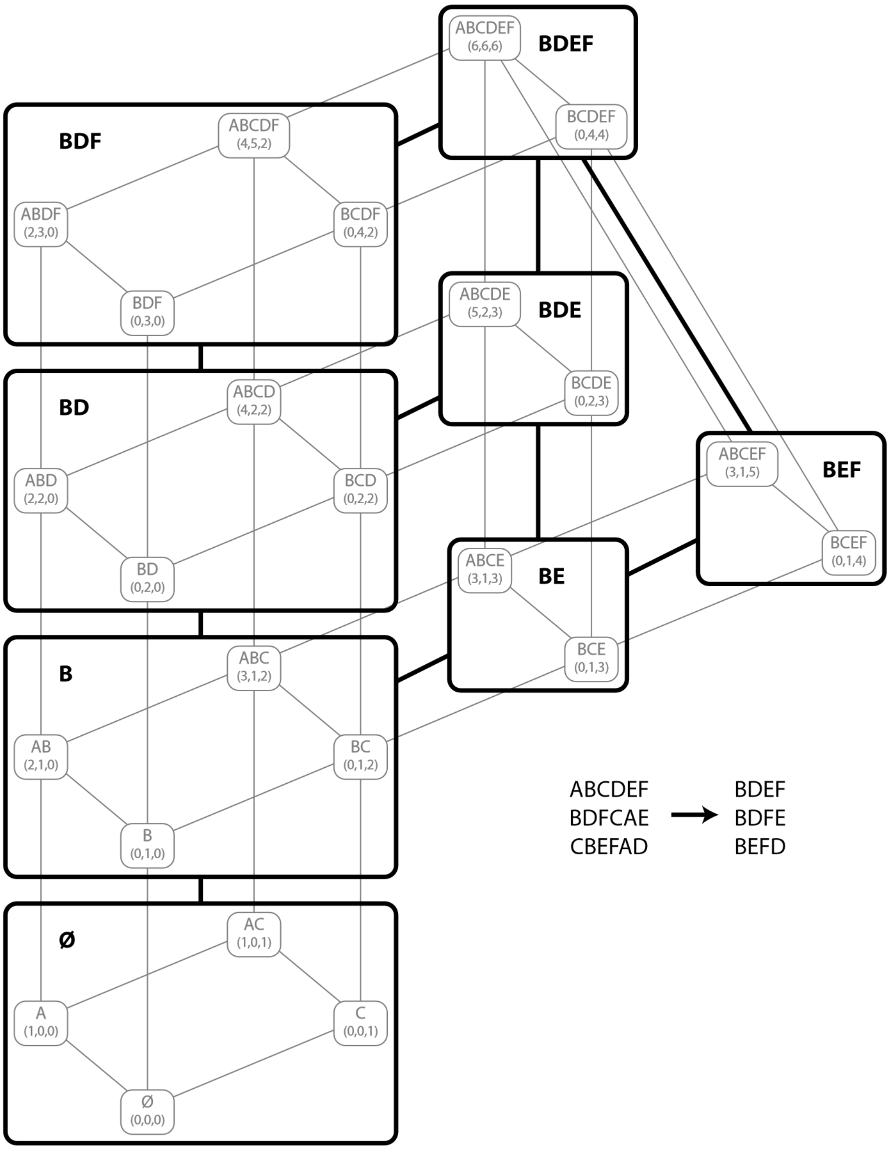

Figure 5.1

In the figure, the states of the original knowledge space is shown in light grey colour while the states of the projection are given in the large black rounded rectangles. Here, ABCDEF, BDFCAE and CBEFAD are the three learning sequences that define the original knowledge space. The corresponding three learning sequences in the projection are BDEF, BDFE and BEFD.

5.2 Fibers of Projection

5.2.1 Definition

The sample of concepts is partitioned into two subsets, a set *L* of concepts that the student knows and a set *U* of concepts that the student does not know. Thus, we can denote the sample as K(*L, U*). Now, we define a projection K_{LUU} from K(*L, U*) onto L. Then, K(*L, U*) is called the fibre of the projection and also it is the inverse image of *L*.

The unique maximal state of K(L, U) denoted by $\neg U$ is formed as the union of a set of prefixes of the learning sequences defining K and it does not depend on L. The set L is a valid state of the projection $K_{L\cup U}$ iff $L \subset \neg U$.

Note

Suppose x does not belong to $L \cup U$.

- If x does not belong to $\neg U$, then there does not exist any states of K such that the student knows $L \cup \{x\}$ and does not know U. Thus, the student does not know x.
- If L is not contained in ($\neg U \cup \{x\}$), then there does not exist any states of K such that the student knows L and does not know $U \cup \{x\}$. Thus, the student knows x.
- There are states of K such that the student knows $L \cup \{x\}$ and does not know U and also there are states such that the student knows L and does not know $U \cup \{x\}$. In this case, we can make no assessment of whether the student knows x .

5.2.2 Example

Consider the knowledge space K of figure 5.2.

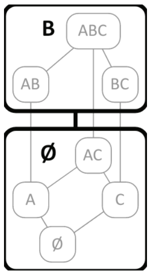

Figure 5.2

Here the projection K$\{B\}$ has two states, $\varnothing$ and $\{B\}$. But, the inverse image K(B,$\varnothing$) of $\{B\}$ is the family of three states $\{\{A,B\},\{A,B,C\},\{B,C\}\}$ which does not form a knowledge space as it fails the accessibility property.

5.2.3 Definition

Let K be a knowledge space. Then the upper subfamily K^+ of K is defined as a subset of K such that if $S \subset T$ are two states of K with $S \in K^+$, then $T \in K^+$.

5.2.4 Theorem

Every fiber K(L, U) forms an upper subfamily of a knowledge space K' and every upper subfamily K^+ of a knowledge space K can be represented as a fiber K'(L,U) for some knowledge space K' and sets L and U.

Proof

Let K be a knowledge space.

Suppose K(L, U) is a fiber and $S = \cup$K(L, U).

Then, S is the unique maximal state of K(L, U) and $S \subset \cup$K.

Consider the projection $K_S = \{P \cap S \mid P \in K\}$

By theorem 5.1.1, K_S is a knowledge space.
Let $A \subset B$ be two states of K_S with $A \in K(L, U)$.
Since $B \in K_S$, $B \in P \cap S$ where $P \in K$
i.e., $B \in K$ and $B \in S$
i.e., $B \in \cup K(L, U)$
i.e., $B \in K(L, U)$ for some partition
Thus, $K(L, U)$ forms an upper subfamily of the knowledge space K_S.
Hence, every fiber $K(L, U)$ forms an upper subfamily of a knowledge space K'
Suppose K^+ is an upper subfamily of a knowledge space K.
Form K' by adding the sets of the form $S \cup \{x\}$ to K where $S \in K^+$ and x not in $\cup K$.
i.e., $K' = K^+ \cup \{x\}$ where x not in $\cup K$.
Now project K' onto $\{x\}$.
Then, K^+ is the inverse image of $\{x\}$ under this projection.
i.e., K^+ is the fiber of this projection which can be represented as $K'(K^+, \{x\})$.
Thus, every upper subfamily K^+ of a knowledge space K can be represented as a fiber $K'(L, U)$ for some knowledge space K' and sets L and U.

5.2.5 Example

Figure 5.3 shows a well-graded, union-closed family F that cannot be an upper subfamily K^+ of a knowledge space K.

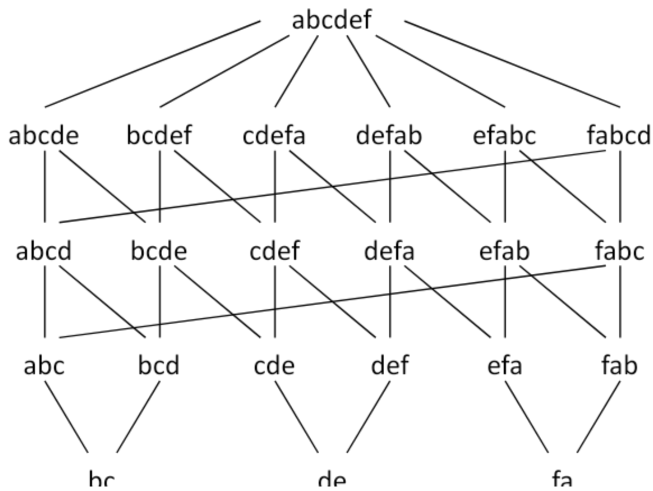

Figure 5.3

Consider the two states $\{d, e\}$ and $\{a, d, e\}$ of K.
Clearly, $\{d, e\} \subset \{a, d, e\}$ and $\{d, e\} \in F$.
But, $\{a, d, e\}$ does not belong to F.
Thus, F is not an upper subfamily of K.

5.3 Chain of Knowledge states

In this section, we discuss a method for finding the smallest set of learning sequences that define a given knowledge space. In this method, an optimal set of learning sequences is constructed using a graph matching algorithm.

5.3.1 Definition

A chain in a partial order is a set of items that are all comparable to each other or equivalently a sequence x_0, x_1, ... such that $x_i < x_j$ iff $i < j$.

A chain cover is a set of chains that together include all items in the order. An antichain is a set of items in which no two of them are comparable to each other.

The width of a partial order is the maximum cardinality of any of its antichains or equivalently the minimum number of chains in a chain cover.

5.3.2 Optimal chain decomposition

An optimal chain decomposition of a given partial order can be found in polynomial time using a bipartite graph matching as illustrated in the figure 5.4.

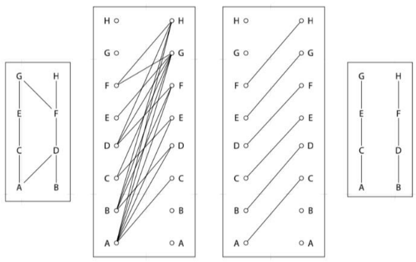

Figure 5.4

The far left of the figure shows a partial order and the center left shows a bipartite graph in which each item in the partial order is represented by two vertices one on each side. The graph has an edge from a vertex x on the left to a vertex y on the right whenever $x < y$. A maximum matching in this graph is shown in the center right. From any matching in this graph, a chain cover can be derived in which the consecutive pairs of elements in the chains are given by the matched edges. The cover derived in this way is shown on the far right of the figure.

In a partial order with n items, matchings with k matched edges correspond to chain covers with $n - k$ chains. So, the minimum chain cover can be found from the maximum matching and the number of chains in it gives the width. If we partially order the states of a knowledge space by set inclusion, then a chain of states consists of a nested family of sets.

5.3.3 Lemma

Let C be a chain of states in a knowledge space K. Then, there exists a learning sequence σ of K such that each state of C is a prefix of σ. If K is defined from a set Σ of learning sequences, then σ can be constructed from C in time polynomial in the total size of Σ.

Proof

Without loss of generality, assume that C contains $\varnothing$ and $\cup$K.

Now, sort the states in C according to its size as follows:

$\varnothing = S0 \subset S1 \subset S2 \subset \cdots \subset Sk = \cup K$.

We construct σ in stages in such a way that after the ith stage, we will choose values for $\sigma(0)$, $\sigma(1)$, ..., $\sigma(|S_i| - 1)$ such that all states S_j, $j \leq i$, are prefixes of σ .

Initially, i = 0 and σ is an empty sequence.

In stage i, a sequence is generated for the elements of S_{i-1}.
Repeatedly applying the accessibility property of knowledge spaces to S_i, we can generate a sequence τ (0), τ (1), ..., τ ($|S_i| - 1$) of the elements of S_i, such that each prefix of this sequence is a state in K.
Then, we concatenate the sequence of σ_j values with the subsequence of τ_j values such that none of them is already chosen as σ_j's.
This concatenation preserves the property that the sets S_j, $j \leq i$ are all prefixes of the new sequence,.
Moreover, each prefix of the new sequence is the union of a prefix of the old sequence of σ_j's and of a prefix of the sequence of τ_j's.
By the union closure property of knowledge spaces, these prefixes are states of K.
After completing the final stage of this process for $S_k = \cup$K, we get the desired learning sequence σ.

5.3.4 Definition

The path poset of a knowledge space is the set of its atoms, partially ordered by inclusion.

5.3.5 Lemma

Let Σ be a (non-optimal) set of learning sequences defining knowledge space K. Then, in time polynomial in the total size of Σ, we can construct the path poset of K.

<u>Proof</u>

By theorem 4.3.9, each atom of K must be a prefix of Σ.
By the definition of atoms, an atom of a knowledge space is a nonempty state S for which there exists only one concept $x \in S$ such that $S \setminus \{x\}$ is a valid state.
Thus, we can determine which prefixes are atoms by computing their inner fringes.
After finding the atoms, we can form a partially ordered set from them, by testing each pair of states for set inclusion. This gives the path poset of K.

5.3.6 Theorem

Let Σ be a (non-optimal) set of learning sequences defining knowledge space K. Then, in time polynomial in the total size of Σ, we can construct a minimum-cardinality set Σ of learning sequences that also defines K.

<u>Proof</u>

By lemma 5.3.5, we can construct the path poset of K.
Now, decompose it into a minimum number of chains by using the bipartite matching algorithm and construct a learning sequence for each chain by Lemma 5.3.3.
Then, every representation of K by these learning sequences will be a chain cover of the path poset.
Since the path poset is decomposed into a minimum number of chains, each representation of K has the minimum possible number of learning sequences.
This gives a minimum-cardinality set Σ of learning sequences that defines K.

5.4 Decomposition of Knowledge Spaces

The problem of finding the minimum cardinality set of learning sequences defining a knowledge space is a form of decomposition of the space.

5.4.1 Definition

If L_1 and L_2 are two knowledge spaces on the same set of concepts, then a join operation combining the two into a single larger knowledge space is defined as:
$L_1 \sqcup \mathcal{L}_2 = \{S_1 \cup S_2 \mid S_i \in \mathcal{L}_i\}$.
This join operation is commutative, associative, and idempotent (i.e., $L \sqcup \mathcal{L} = L$ for any knowledge space L).
The problem of representing knowledge space K using a minimum number of learning sequences is equivalent to expressing K as the join of a small number of simpler knowledge spaces.

5.4.2 Definition

A totally ordered knowledge space is a knowledge space formed from a single learning sequence.

Result

- K can be represented using k learning sequences iff it is the join of at most k totally ordered knowledge spaces.
- The totally ordered knowledge spaces form the smallest subclass of knowledge spaces such that all knowledge spaces can be represented as joins of totally ordered knowledge spaces.
- A knowledge space is irreducible iff it is totally ordered.

5.4.3 Theorem

It is NP-complete, given a set B of atoms that define a knowledge space K and given an integer *k*, to determine whether K is the join of *k* or fewer quasi-ordinal spaces.

5.5 Adapting a learning sequence

If K is a knowledge space, then the inner fringe of K consists of those states of K that may be removed, leaving a knowledge space for the same set of concepts with fewer states while the outer fringe of K consists of those sets that are not states of K but may be added as states, resulting in a knowledge space on the same set of concepts but with more states. In this section, we describe how to calculate the fringes of a knowledge space defined from a set of learning sequences.

5.5.1 Two fringes of a knowledge state

Let *S* be a knowledge state in the knowledge space K. Then its outer fringe consists of the concepts *x* which does not belong to *S* such that $S \cup \{x\}$ is also a state and its inner fringe consists of the concepts $x \in S$ such that $S \setminus \{x\}$ is also a state. The fringe of *S* is the union of its outer and inner fringes. i.e, the fringe of *S* is the set of concepts that, when added to or removed from *S*, lead to another state in K.

5.5.2 Theorem

A state *S* belongs to the inner fringe of K iff it satisfies the following conditions:

1. *S* is an atom of K.
2. *S* not equal to $\cup$K.

3. No set $S \cup \{x\}$ where x does not belongs to S, is an atom of K.

Proof

Necessary Part

Suppose S belongs to the inner fringe of K.
1.Assume S is not an atom.
Then, S could be formed as the union of other sets in K.
Now, remove S from K.
Then, the union of sets in K does not belong to K which violates the union- closure property of K.
Thus, S is an atom of K.
2.Suppose S is an atom but equals $\cup$K.
Now, remove S from K.
Then, $\cup$K does not belongs to K which violates the requirement that the resulting space have the same set of concepts.
Hence, S not equal to $\cup$K.
3. Assume $S \cup \{x\}$ where x not in S is an atom for some x.
By accessibility property of $S \cup \{x\}$, $S \cup \{x\} \backslash x = S \in$ K
Removal of S would violate accessibility property of $S \cup \{x\}$.
Thus, no set $S \cup \{x\}$ where x does not belongs to S, is an atom of K.

Sufficient Part

Suppose the conditions 1,2,3 holds.
Since S is an atom, it cannot be the union of other sets.
So, removal of S does not affect the union-closure property of K.
Since S is not equal to $\cup$K of K, its removal do not change the set of concepts of the resulting space.
Because each state $S \cup \{x\}$ is not an atom, it has at least two removable elements with at least one of which is not x.
Then, the removal of S preserves the accessibility property of K.
Hence, removal of S forms a knowledge space on the same set of concepts, meeting the definition of the inner fringe.

5.5.3 Theorem

A set T belongs to the outer fringe of K iff it has the form $S \cup \{x\}$ where S is a state of K and x belongs to the intersection of the outer fringes of all states of the form $S \cup \{y\}$.

Proof

Necessary Part

Suppose T belongs to the outer fringe of K.
Assume T does not have the form $S \cup \{x\}$ where S is a state of K.
Consider $S \cup T$.

Let $z \in S \cup T$
Clearly, $S \cup T \setminus z$ does not belong to K since T contains at least two elements which are not in K.
Thus, accessibility property does not holds for $S \cup T$.
So, the set system formed by adding T will not be accessible.
Hence, T has the form $S \cup \{x\}$ where S is a state of K.
Suppose there exists a state $S \cup \{y\}$ in K such that x does not belong to its outer fringe.
We have proved that T has the form $S \cup \{x\}$ where S is a state of K.
Then, $T \cup (S \cup \{y\}) = (S \cup \{x\}) \cup (S \cup \{y\})$ which does not belong to K.
i.e., union-closure property does not holds for $T \cup (S \cup \{y\})$.
So, the set system formed by adding T is not closed under unions.
Hence, x belongs to the intersection of the outer fringes of all states of the form $S \cup \{y\}$.

Sufficient Part

Suppose $T = S \cup \{x\}$ where S is a state of K and x belongs to the intersection of the outer fringes of all states of the form $S \cup \{y\}$.
Clearly, the set system formed by adding T will be accessible.
Let U be a state of the form $S \cup \{y\}$.
Case 1: If $U \subset S$
Then, $U \cup T = U \cup (S \cup \{x\})$
$= S \cup \{x\} = T$
Case 2: If U is not contained in S

Let y $\in U \setminus S$
Then, $S \cup \{y\} \in$ K.
Here, $U \cup T = (S \cup \{y\}) \cup T$ where $S \cup \{y\} \in$ K.
Thus, in both cases, the set system formed by adding T is closed under union.
Hence, the set system formed by adding T is a knowledge space.
Therefore, T belongs to the outer fringe of K.

5.6 APPLICATIONS

Some of the first applications of knowledge space theory were in mathematical education. It can be used to represent students' arithmetic knowledge in a test that assesses fractions and decimal numbers. Also, it is helpful to represent students' knowledge of algebra.

In science education, KST can be applied for the evaluation of an instructional strategy for teaching the concepts of pressure, density and the conservation of matter. Thus, it helps to identify the most probable learning pathways and to evaluate the efficiency of the applied instructional strategies.

The KST framework has allowed researchers to identify the students' critical learning pathways as well as to compare them with experts' critical learning pathways. Some important applications of KST in science education include its combination with other methodologies, such as phenomenography with the aim of investigating students' reasoning and changes in cognitive structures.

CONCLUSION

Knowledge space theory (KST) is a mathematical method for modelling students' knowledge. There are many applications for the skill-based extension of knowledge space theory. If the items in the knowledge domain are not fixed, then it is possible to include additional items. It is done by relating the new item to all the old ones in order to localize it within the knowledge structure. For this, identify the relevant skills needed to solve the existing items. Then, assign the relevant competencies to the new item. Once this is done, the relationship of the new item to the old ones is uniquely determined. This property is essential for any application of knowledge space theory to technology-enhanced learning within an open systems architecture, where the item repositories are not static but may be modified by adding or removing items.

The main applications of competence based KST are in technology enhanced learning (TEL) such as in general personalized TEL, in non-invasive TEL for game- and simulation-based learning and in competence management and workplace learning. Competence based KST also has applications in developmental psychology, philosophy for children etc.

We have learned how to form a sample from a knowledge space and to form a projected space and recognize the fibers of this projection. Also, we have studied how to represent knowledge spaces using learning sequences and to decompose knowledge spaces into small numbers of simpler spaces. Also learnt some algorithms for listing their respective states and learning sequence representations. These methods are useful while considering very large knowledge spaces.

REFERENCE

1. Falmagne, J.-Cl.; Albert, D; Doble, C.W; Eppstein, D; and Hu, X. 2013. *Knowledge Spaces: Applications in Education*. Berlin: Springer.
2. Doignon, J.-P; and Falmagne, J.-Cl. 1999. *Knowledge Spaces*. Berlin: Springer.
3. Doignon, J.-P; and Falmagne, J.-Cl. 2015. Knowledge spaces and learningspaces.URL:https://www.researchgate.net/publication/284579542