國 立 交 通 大 學

資訊管理研究所

博　士　論　文

決 策 邏 輯 型 機 制 及 其 在 知 識 表 徵 中 之 應 用
Decision Logic-Styled Formalisms for Knowledge Representation

研 究 生： 范端芳

指導教授： 劉敦仁 博士

中華民國 97 年 7 月

決策邏輯型機制及其在知識表徵中之應用
Decision Logic-Styled Formalisms for Knowledge Representation

研 究 生： 范端芳　　Student: Tuan-Fang Fan

指導教授： 劉敦仁　　Advisor: Duen-Ren Liu

國立交通大學

資訊管理研究所

博士論文

A Dissertation

Submitted to Institute of Information Management

College of Management

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy in Information Management

July 2008

Hsinchu, Taiwan, the Republic of China

中華民國 97 年 7 月

致謝

一篇博士論文的完成，一定有許多需要感謝的人，我的也不例外。首先要感謝的當然是指導教授劉敦仁博士在各方面的指導與協助。另外，曾國雄教授在多目標決策分析課程上給我的啟發，是本論文的重要起點。同時也要感謝各位口試委員提供的寶貴意見，使論文無論是在內容或格式上都能有所改進。另外，同學楊耿杰先生在我計畫書與論文兩場口試時，提供的各種協助，使得口試能夠順利進行，在此也要一併致謝。最後，家人與小犬給我的精神支持，是我遇到困難時能夠一一加以克服的重要關鍵，我要特別謝謝他（它）們。

摘要

近年來，從資料庫中發掘知識與其核心機制—資料探勘越來越受到廣泛的注意。雖然資料探勘研究一貫是以設計高效率的演算法為主，然如何使探勘所得的知識能對使用者有用，仍然持續成為該領域一個最具挑戰性的問題。由於知識能對使用者有用的先決條件是使用者能瞭解其意義，因此知識表徵機制在知識管理過程中便扮演著重要的角色。

我們在本學位論文中探討的就是從粗略集合論觀點對決策邏輯作若干擴充。傳統決策邏輯是以粗略集合論為基礎的資料探勘中一種標準的知識表徵機制，而我們的擴充顯示出決策邏輯型機制對於較複雜的知識管理工作亦非常有用。

我們一方面提出數種決策邏輯語言，可用於表達以粗略集合論為基礎的多準則決策分析方法所產生的決策法則，這些語言的語意模型為表示多準則決策記錄之資料表，其中每一個決策記錄都可以用有限多個屬性或準則來描述。而準則與屬性的差別是屬性值之間不見得有優劣關係，而準則的值之間必然存在優劣關係。

另一方面，我們提出矢決策邏輯來對關聯資訊系統中所發掘出來的知識作表徵與推理。此一邏輯係結合矢邏輯與決策邏輯的主要特徵而成，其中矢邏輯為一種用於關係推理的樣態邏輯。矢決策邏輯的邏輯式可以在關聯資訊系統中加以解釋，而關聯資訊系統不僅描述物件的屬性，亦描述其彼此之間的關係。我們提出一種矢決策邏輯的公設化系統，證明其完備性，並展現其在多準則決策分析與社交網路分析上的應用。

我們的結果對知識管理中知識表徵此一環節特別有用。我們以一個現實的例子來說明我們所提出來的機制可用來輔助公司聘僱人員及形成團隊過程中不同階段的知識表徵需求。

Abstract

In recent years, knowledge discovery in databases (KDD) and its kernel data mining have received more and more attention for practical applications. While the mainstream research of data mining concentrates on the design of efficient algorithms for extracting knowledge from databases, the question to close the semantic gap between structured data and human-comprehensible concepts has been a lasting challenge for the research community. Since the discovered knowledge is useful for a human user only when he can understand its meaning, the representation formalism will play an important role during the knowledge management life cycle.

In this dissertation, we investigate several extensions of decision logic (DL) from the perspective of rough set theory. Traditionally, DL has been considered as a standard way of knowledge representation for rough set-based data mining, whereas our extensions show that DL-styled logics are also useful in more complicated knowledge management tasks.

On the one hand, we propose some decision logic languages for rule representation in rough set-based multicriteria decision analysis. The semantic models of these logics are data tables representing multicriteria decision records. Each decision record is described by a finite set of criteria/attributes. The domains of the criteria may have ordinal properties expressing preference scales, while the domains of the attributes may not.

On the other hand, we propose an arrow decision logic (ADL) to represent and reason about knowledge discovered from relational information systems (RIS). The logic combines the main features of decision logic (DL) and arrow logic (AL). AL is the basic modal logic of arrows. ADL formulas are interpreted in RIS which not only specifies the properties of objects, but also the relationships between objects. We present a complete axiomatization of ADL and discuss its application to knowledge representation in multicriteria decision analysis and social network analysis.

Our work is particularly useful for the knowledge representation phase in the knowledge management life cycle. A realistic scenario about human resource management is used to show how the proposed logics can serve as representational formalisms in different stages of the recruitment process and team formation process of a company.

Contents

1 Introduction **1**
- 1.1 Outline of the Thesis . . . 3
- 1.2 The Main Theme and Contributions . . . 4

2 Information Technology and Knowledge Management **6**
- 2.1 Rough Set Theory . . . 6
  - 2.1.1 Approximation space . . . 6
  - 2.1.2 Data tables and decision logic . . . 7
  - 2.1.3 The connection . . . 10
- 2.2 Logical Preliminary . . . 10
  - 2.2.1 Propositional logic . . . 10
  - 2.2.2 Modal logic . . . 11
  - 2.2.3 Arrow logic (AL) . . . 12
- 2.3 Knowledge Management . . . 14
  - 2.3.1 Knowledge management process . . . 14
  - 2.3.2 Logic-based knowledge management . . . 18
  - 2.3.3 A running scenario . . . 18
- 2.4 Related Work . . . 20

3 Decision Logics for Multicriteria Decision Analysis **22**
- 3.1 Preference-ordered Data Tables . . . 23
  - 3.1.1 The running scenario . . . 24
- 3.2 Preference-ordered Uncertain Data Tables . . . 26
  - 3.2.1 The running scenario . . . 31
- 3.3 Preference-ordered Fuzzy Data Tables . . . 32
  - 3.3.1 The running scenario . . . 37
- 3.4 Pairwise Comparison Decision Logic . . . 38
  - 3.4.1 The running scenario . . . 40
- 3.5 Summary and Remarks . . . 43

4 Arrow Decision Logic for Relational Information Systems **44**
- 4.1 Information Systems . . . 45
  - 4.1.1 Functional and relational information systems . . . 45
  - 4.1.2 Relationship between information systems . . . 46
- 4.2 General Theory of Granulation . . . 51

4.3 Arrow Decision Logic 53
4.3.1 Syntax and semantics of ADL 53
4.3.2 Axiomatization 55
4.4 Discussion and Applications 57
4.4.1 Discussion 57
4.4.2 An Application of ADL to MCDM 58
4.4.3 An application of ADL to the representation of attribute dependency . . . 60
4.4.4 The running scenario 61
4.5 Summary and Remarks 61

5 Conclusion **66**
5.1 The Contribution of the Work 66
5.2 Remarks 66
5.2.1 A philosophical consideration 67
5.2.2 Hypothesis selection 70
5.3 Future Research 72

List of Figures

1.1 An interface is needed between the KDD and AI systems 2
1.2 The organization of the paper 4

2.1 An arrow model 14
2.2 The knowledge management life cycle 17
2.3 The preliminary classification phase of the recruitment process 21

3.1 The short-term performance prediction phase of the recruitment process 25
3.2 The long-term performance prediction phase of the recruitment process 33
3.3 The membership functions of fuzzy sets 35
3.4 The project proposal test phase of the recruitment process 38
3.5 The comparison of possible scores $s(x)$ and $s(y)$ 41
3.6 The final decision phase of the recruitment process 42

4.1 An ADL model 55
4.2 The axiomatic system for ADL 63
4.3 The MCDA and MCDM phases 64
4.4 The process of team formation 65

5.1 The process of the running scenario 67

List of Tables

2.1 A summary of reviewers' reports for 10 papers 9

3.1 A POUDT for route selection 31
3.2 A data table for project evaluation 36

4.1 An RIS obtained from a given embedding 51

Chapter 1

Introduction

Knowledge management (KM) is a discipline concerned with theories, tools and methodologies that can help individuals, groups and organizations manage the knowledge they use on a daily basis. The theory of knowledge has long been an important topic in many academic disciplines, such as philosophy, psychology, economics, and artificial intelligence, whereas the storage and retrieval of data is the main concern of information science. From a KM perspective, knowledge is usually acquired from observed data, which is a valuable resource for researchers and decision-makers. However, when the amount of data is large, it is difficult to analyze the data and extract knowledge from it. With the aid of computers, the vast amount of data stored in relational data tables can be transformed into symbolic knowledge automatically. Thus, intelligent data analysis has received a great deal of attention of KM researchers in recent years.

Especially, knowledge discovery in databases (KDD) and its kernel data mining have received more and more attention for practical applications. While the mainstream research of data mining concentrates on the design of efficient algorithms for extracting knowledge from databases, the question to close the semantic gap between structured data and human-comprehensible concepts has been a lasting challenge for the research community [50]. This is called the interpretability problem of intelligent data analysis in [50]. Since the discovered knowledge is useful for a human user only when he can understand its meaning, the representation formalism will play an important role during the KM life cycle. As mentioned in [72], the key open problem of knowledge organization in the KM life cycle is "knowledge representation using a universal language that may allow multiple channel and experience support".

Many different forms of knowledge have been considered by the KDD researchers, notably, the association rules and sequential patterns [1, 2]. However, it is in general difficult to integrate the discovered patterns and traditional AI systems. The main reason is that the inference engine of AI systems usually employ a logic-based knowledge representation, which is quite different from the specialized patterns discovered by a fixed data mining algorithm. Therefore, a uniform interface between the discovery and utilization of knowledge is urgently needed. The interface will transform the discovered patterns into the knowledge based on the logical formalism employed by the AI system (Figure 1.1). From the KM viewpoint, the KDD process is concerned with knowledge creation, whereas the AI system is designed for knowledge application. Thus, the smooth interface between these two will improve the coherence of the whole KM process.

The advantages of the logic-based representation for data mining have also been observed in the past [26].

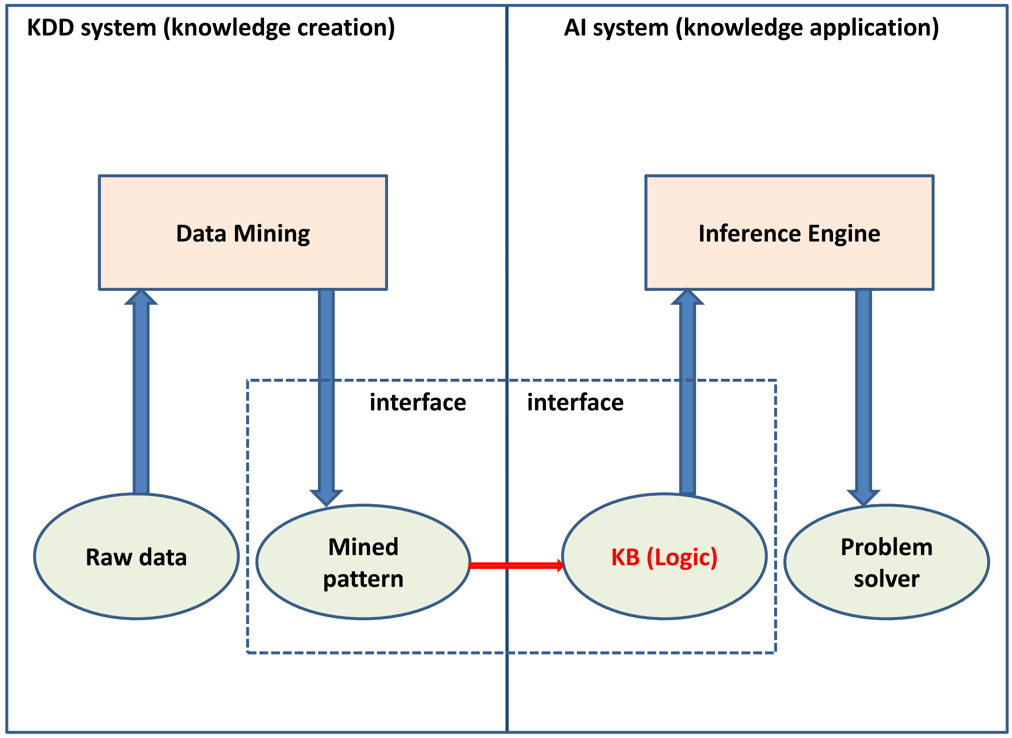


Figure 1.1: An interface is needed between the KDD and AI systems

... a coherent formalism, capable of dealing uniformly with induced knowledge and background, or domain, knowledge, would represent a breakthrough in the design and development of decision support systems, in diverse application domains. The advantages of such an integrated formalism are, in principle:

- *a high degree of expressiveness in specifying expert rules, or business rules;*
- *the ability to formalize the overall KDD process, thus tailoring a methodology to a specific class of applications;*
- *the separation of concerns between the specification level and the mapping to the underlying databases and data mining tools.*

The rough set theory proposed by Pawlak provides an effective tool for extracting knowledge from data tables [67]. In fact, many powerful data mining algorithms have been proposed based on the rough set theory (for example, see papers in [75, 76, 68] for some recent progress). To represent and reason about the extracted knowledge, a decision logic (DL) is also proposed in [67]. The semantics of the logic is defined in a Tarskian style through the notions of models and satisfaction.

Due to the following two reasons, DL is a good candidate to serve as the bridge between the KDD and AI systems: On the one hand, the data mining algorithms based on rough set theory usually extract rules which can be easily represented in the syntactical form of DL language. On the other hand, the semantic similarity between DL and Classical logic makes it easier to integrate the mined results into knowledge-based systems.

Although DL can faithfully represent knowledge discovered from classical data tables, it is inadequate for more advanced data mining tasks, such as uncertainty management or multi-criteria decision analysis. To address such challenging issues, this work is aimed at the investigation of different extensions of DL that can represent more complicated forms of knowledge. Though we can envision a powerful universal logic that integrates all features of the extended DL languages, it is impractical to devise a logic that can cover all possible forms of knowledge. Thus, our strategy is to develop a particular extended DL for each individual aspect of the knowledge representation requirements of different data mining tasks, while keep it easy to modularly combine these different extended logics.

1.1 Outline of the Thesis

One key step in knowledge management is the transformation of data and information into knowledge. Data mining can play an important role in such a transformation process. To set up the foundations of the dissertation, we review rough set theory, which is an effective method of data mining; propositional and modal logics, which are basic knowledge representation formalisms; and the theory of knowledge management, which provides an appropriate context of our work.

In classical data tables, no relationship exists between values in a domain of attribute. However, in many cases, in particular, when the attribute is a criterion for decision making, there exist several preference relations on the domain. When rough set theory is applied to multi-criteria decision analysis (MCDA), it is crucial that preference-ordered attribute domains and decision classes be dealt with [29, 30, 31, 33, 34, 35, 36, 82]. The original rough set theory cannot handle inconsistencies arising from violations of the dominance principle due to its use of the indiscernibility relation. Therefore, in the above-mentioned work, the indiscernibility relation is replaced by a dominance relation to solve the multi-criteria sorting problem, and the data table is replaced by a pairwise comparison table to solve multi-criteria choice and ranking problems. The approach is called the dominance-based rough set approach (DRSA). For MCDA problems, DRSA can induce a set of decision rules from sample decisions provided by decision-makers. The induced rules form a comprehensive preference model and can provide recommendations about a new decision-making environment. The objective of chapter 3 is to investigate DL for DRSA.

In addition of preference relations, general relations may exist on the domains of attribute values. These relations may induce complicated relationship between objects possessing the attributes. For example, a social network is a set of objects with complicated interaction relations. To model such kind of data, we must consider relational information systems (RIS). In data tables, each attribute is actually a function from the set of objects to the domain of attribute values, so a data table is also called a functional information system (FIS). In contrast with FIS, an attribute in a RIS is considered as a function from a pair of objects to the domain of relational indicators. The objective of Chapter 4 is to develop a logic for relational information systems.

The materials in this thesis are mainly drawn from some of our previously published papers. Chapter 3 is a slightly adapted version of [21]. Chapter 4 is a fusion of [20] and [18]. Finally, Chapter 5 contains partially the contents of [17] and [19]. The organization of the thesis is shown in Figure 1.2.

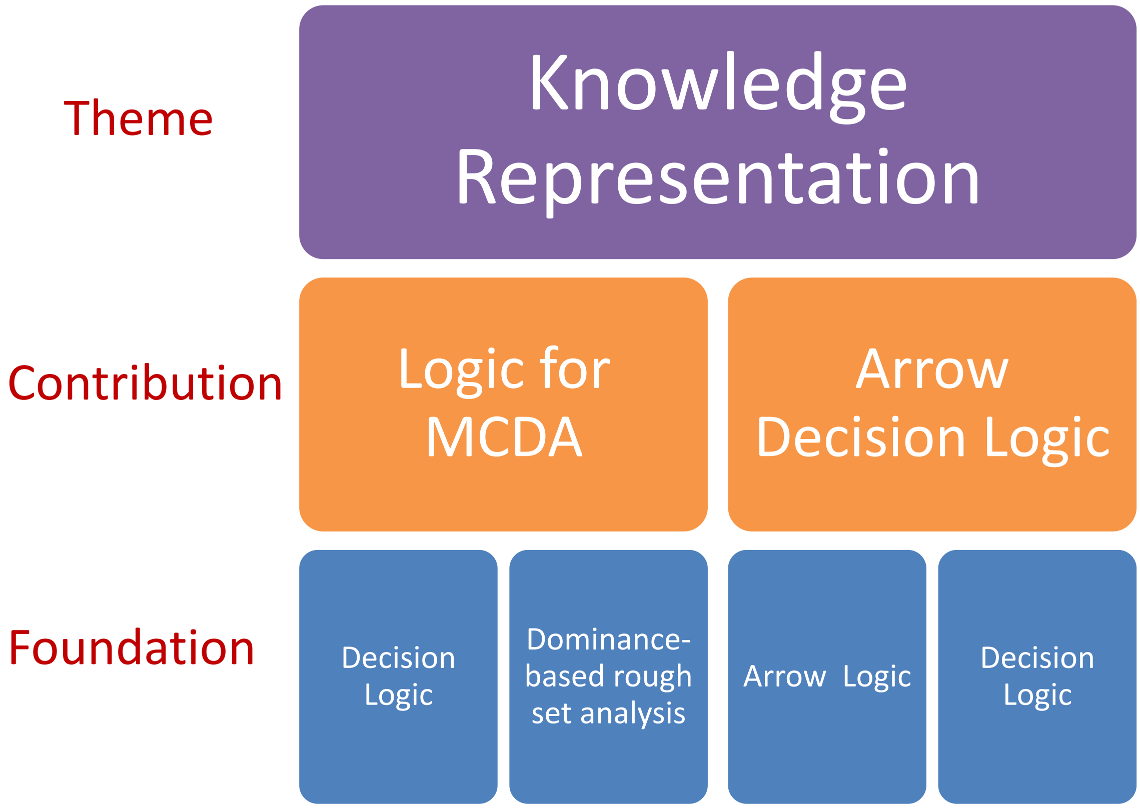


Figure 1.2: The organization of the paper

1.2 The Main Theme and Contributions

From Figure 1.2, we can see that the main theme of our work is to design knowledge representation formalisms in a uniform style. The foundations of our knowledge representation formalisms include decision logic, arrow logic, and (dominance-based) rough set analysis. As a consequence, we have proposed different extensions of decision logics that can accommodate knowledge representation requirements in different knowledge management tasks. Thus, the main contributions of the thesis are the development of the following extensions of decision logics for representing different forms of knowledge:

- PODL: for knowledge from the data of preference-ordered domains (e.g., evaluation data),
- POUDL: for knowledge discovered from data tables where data is uncertain or information is incomplete,
- POFDL: for knowledge discovered from data tables where data is qualitatively and imprecise,
- PCDL: for knowledge discovered from pairwise comparisons of objects, and

- ADL: for knowledge discovered from relational information systems (e.g., social network data).

Chapter 2

Information Technology and Knowledge Management

In this chapter, we introduce necessary background knowledge for our work. Furthermore, we also mention several related work in the last section.

2.1 Rough Set Theory

2.1.1 Approximation space

The basic construct of rough set theory is an *approximation space*. An approximation space is defined as a pair (U, R), where U is the set of universe and $R \subseteq U \times U$ is an equivalence relation on U. Recall that a binary relation R is an equivalence relation if it is reflexive (i.e., $(x, x) \in R$ for all $x \in U$), symmetric (i.e., for all $x, y \in U$, if $(x, y) \in R$, then $(y, x) \in R$), and transitive (i.e., for all $x, y, z \in U$, if $(x, y) \in R$ and $(y, z) \in R$, then $(x, z) \in R$). An equivalence relation partitions the universe U into a family of equivalence classes so that each element of U belongs to exactly one of these equivalence classes. In other words, there exist $U_1, U_2, \cdots, U_k \subseteq U$ such that $U = \cup_{i=1}^{k} U_i$, $U_i \cap U_j = \emptyset$ for $i \neq j$, and for $x, y \in U$, $(x, y) \in R$ if and only if (iff) there exists i such that both x and $y \in U_i$. Thus, we can write an equivalence class of R as $[x]_R$ if it contains the element x. Note that $[x]_R = [y]_R$ iff $(x, y) \in R$.

According to Pawlak's insight, it is claimed that knowledge is deep-seated in the classification capabilities of human beings. A classification is simply a partition of the universe, so an approximation space construct the basic knowledge about the objects in the universe. In philosophy, the extension of a concept is defined as the objects that are the instances of the concept. For example, the extension of the concept "bird" is simply the set of all birds in the universe. Pawlak identified a concept with its extension. Thus, a subset of the universe is called a *concept* or a *category* in rough set theory.

Given an approximation space (U, R), each equivalence class of R is called a *R-basic category* or *R-basic concept*, and any union of R-basic categories is called a R-category. Now, for an arbitrary concept $X \subseteq U$, we are interested in the definability of X by using R-basic categories. We say that X is *R-definable*, if X is a R-category; otherwise X is *R-undefinable*. The R-definable concepts are also called *R-exact* sets, whereas R-undefinable concepts are said to be *R-inexact* or *R-rough*.

When the approximation space is explicit from the context, we simply omit the qualifier R and call a set exact set or rough set.

A rough set can be approximated from lower and upper by two exact sets. The lower approximation and upper approximation of X is denoted by $\underline{R}X$ and $\overline{R}X$ and defined as follows respectively:

$$\underline{R}X = \{x \in U \mid [x]_R \subseteq X\},$$
$$\overline{R}X = \{x \in U \mid [x]_R \cap X \neq \emptyset\}.$$

2.1.2 Data tables and decision logic

In data mining problems, data is usually provided in the form of data tables (DT). A formal definition of a data table is given in [67].

Definition 2.1 *A data table*[1] *is a tuple*

$$T = (U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\}),$$

where U is a nonempty finite set, called the universe; A is a nonempty finite set of primitive attributes; for each $i \in A$, V_i is the domain of values for i; and for each $i \in A$, $f_i : U \rightarrow V_i$ is a total function.

Given a data table T, we denote its universe U and attribute set A by $Uni(T)$ and $Att(T)$ respectively. An attribute in A is usually denoted by the lower-case letters i or a.

In [67], a decision logic (DL) is proposed for the representation of knowledge discovered from data tables. It is called decision logic because it is particularly useful in a special kind of data table, called a *decision table.*[2] A decision table is a data table $T = (U, C \cup D, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$, where $Att(T)$ can be partitioned into two sets, C and D, called condition attributes and decision attributes respectively. Decision rules relating the condition and the decision attributes can be derived from the table by data analysis. A rule is then represented as an implication between the formulas of the logic.

The basic alphabet of a DL consists of a finite set of attribute symbols A, and a finite set of value symbols V_i for $i \in \mathsf{A}$. The syntax of DL is then defined as follows.

Definition 2.2

1. *An atomic formula of DL is a descriptor (i, v), where $i \in \mathsf{A}$ and $v \in \mathsf{V}_i$.*

2. *The set of DL well-formed formulas (wff) is the smallest set containing the atomic formulas and closed under the Boolean connectives $\neg, \wedge$, and $\vee$.*

3. *If φ and ψ are wffs of DL, then $\varphi \longrightarrow \psi$ is a rule in DL, where φ is called the antecedent of the rule and ψ the consequent.*

[1] Also called knowledge representation systems, information systems, or attribute-value systems

[2] note that for a general data table, the abbreviation DL can also be used to denote *data logic.*

A data table $T = (U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$ relates to a given DL if there is a bijection $\tau : \mathsf{A} \to A$ such that, for every $a \in \mathsf{A}$, $V_{\tau(a)} = \mathsf{V}_a$. Thus, by somewhat abusing the notation, we usually denote an atomic formula as (i, v), where $i \in A$ and $v \in V_i$ if the data tables are clear from the context. Intuitively, each element in the universe of a data table corresponds to a data record, and an atomic formula (which is in fact an attribute-value pair) describes the value of some attribute in the data record. Thus, the atomic formulas (and therefore the wffs) can be satisfied or not with respect to each data record. This generates a satisfaction relation between the universe and the set of wffs.

Definition 2.3 *Given a DL and a data table $T = (U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$ relating to it, the satisfaction relation $\models_T$ between U and the wffs of the DL is defined inductively as follows (the subscript T is omitted for brevity).*

1. $x \models (i, v)$ *iff* $f_i(x) = v$,
2. $x \models \neg\varphi$ *iff* $x \not\models \varphi$,
3. $x \models \varphi \wedge \psi$ *iff* $x \models \varphi$ *and* $x \models \psi$,
4. $x \models \varphi \vee \psi$ *iff* $x \models \varphi$ *or* $x \models \psi$.

If φ is a DL wff, the set $m_T(\varphi)$ defined by:

$$m_T(\varphi) = \{x \in U \mid x \models \varphi\}, \tag{2.1}$$

is called the meaning set of the formula φ in T. If T is understood, we simply write $m(\varphi)$.

Sometimes, the notations $T, x \models \varphi$ and $x \models_T \varphi$ are considered interchangeable if the data table T must be made explicit.

A formula φ is said to be *valid* in a data table T (written as $\models_T \varphi$ or $\models \varphi$ for short when T is clear from the context) if and only if $m(\varphi) = U$. That is, φ is satisfied by all individuals in the universe. Also, φ is said to be *satisfiable* in a data table T if $m(\varphi) \neq \emptyset$

A DL wff states the properties of individuals in the universe; therefore, it is satisfied by some individuals, but not by the others. However, the mined knowledge usually relates to the aggregated or statistical information of all individuals. Obviously, wffs that are valid in a data table represent a kind of knowledge that can be induced from the table, since they hold for all individuals. However, not all kinds of useful information are in the form of valid wffs. Sometimes, even probabilistic rules are very useful from the viewpoint of knowledge discovery. To quantify the usefulness of the mined rules, some measures have been proposed [95, 93]. The most common measures are support and confidence.

Definition 2.4 *Let Φ_1 be the set of all DL rules and $T = (U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$ be a data table. Then:*

1. *the rule $\varphi \longrightarrow \psi$ is valid in T iff $m_T(\varphi) \subseteq m_T(\psi)$;*
2. *the absolute support function $asp_T : \Phi_1 \to \aleph$ is*

$$asp_T(\varphi \longrightarrow \psi) = |m_T(\varphi \wedge \psi)|;$$

Table 2.1: A summary of reviewers' reports for 10 papers

$U \setminus A$	o	p	t	d
1	4	4	3	4
2	3	2	3	3
3	4	3	2	3
4	2	2	2	2
5	2	1	2	1
6	3	1	2	1
7	3	2	2	2
8	4	1	2	2
9	3	3	2	3
10	4	3	3	3

3. *the relative support function* $rsp_T : \Phi_1 \rightarrow [0,1]$ *is*

$$rsp_T(\varphi \longrightarrow \psi) = \frac{|m_T(\varphi \wedge \psi)|}{|U|}; \text{ and}$$

4. *the confidence function* $cfd_T : \Phi_1 \rightarrow [0,1]$ *is*

$$cfd_T(\varphi \longrightarrow \psi) = \frac{|m_T(\varphi \wedge \psi)|}{|m_T(\varphi)|}.$$

Example 2.1 Let us use an example to illustrate the concept introduced in this section. Assume that Table 1 is a summary of reviewers' report for ten papers submitted to a journal. The table details ten papers evaluated by means of four attributes:

- o: originality,
- p: presentation,
- t: technical soundness, and
- d: overall evaluation (the decision attribute)

By Definition 2.1, the components of the data table are:

$U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$,

$A = \{o, p, t, d\}$,

$V_o = V_p = V_t =$\{1 (poor), 2 (fair), 3 (good), 4 (excellent)\},

$V_d =$\{1 (reject), 2 (major revision), 3 (minor revision), 4 (accept)\},

$f_i(j)(i \in A, 1 \leq j \leq 10)$ denotes the jth element of column i of the table.

Thus, we have atomic formulas like $(o, 4)$, $(p, 1)$, and $(t, 2)$; and formulas like $(o, 4) \wedge (p, 3)$ and $\neg(p, 1) \vee \neg(t, 1)$. The rule $r = (o, 3) \wedge ((p, 3) \vee (t, 3)) \longrightarrow (d, 3)$ is valid, since $m((o, 3) \wedge ((p, 3) \vee (t, 3))) = \{2, 9\} \subseteq m((d, 3)) = \{2, 3, 9, 0\}$. Furthermore, we have $asp(r) = 2$, $rsp(r) = \frac{1}{5}$, and $cfd(r) = 1$. ■

2.1.3 The connection

While an approximation space is an abstract framework to represent classification knowledge, it can be easily derived from a concrete data table. Let $T = (U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$ be a data table and $B \subseteq A$ be a subset of attributes, then we can define an equivalence relation, called the *indiscernibility relation* based on B, as

$$ind(B) = \{(x, y) \mid x, y \in U, f_i(x) = f_i(y) \forall i \in B\}.$$

In other words, x and y are B-indiscernible if they have the same values with respect to all attributes in B. Consequently, for each $B \subseteq A$, $(U, ind(B))$ is an approximation space.

In terms of DL, each equivalence class of B is characterized by a DL formula $\wedge_{i \in B}(i, v_i)$ and any formula φ of DL can be considered as a concept $m_T(\varphi)$. Then, the equivalence class is a subset of the lower (resp. upper) approximation of the concept if the rule $\wedge_{i \in B}(i, v_i) \longrightarrow \varphi$ is valid (resp. the formula $\wedge_{i \in B}(i, v_i) \wedge \varphi$ is satisfiable).

2.2 Logical Preliminary

Logic is commonly defined as the analysis of methods of reasoning[63]. In the presentation of modern symbolic logic, the form of a statement and its content are usually separated, and are defined by the syntax and the semantics of the logic respectively. The syntax of a logic defines a formal language by some grammatical rules and its semantics stipulates the truth conditions of the formulas in the language.

2.2.1 Propositional logic

The syntax of propositional logic (PL) is based on the combinations of simple sentences in various ways to form more complicated sentences. The combinations are *truth-functional* in the sense that the truth value of the new sentence is determined by those of its component sentences. The alphabet of PL consists of a set of primitive propositions, Φ_0, and the logical symbols $\neg$(negation), $\wedge$(and), $\vee$(or), and $\supset$(material implication). The logical symbols of PL are also called *Boolean connectives*. The set of well-formed formulas (wffs) of propositional logic is defined as the smallest set Φ such that $\Phi_0 \subseteq \Phi$ and

- if $\varphi \in \Phi$, then $\neg\varphi \in \Phi$;
- if φ and $\psi \in \Phi$, then $\varphi \wedge \psi, \varphi \vee \psi$, and $\varphi \supset \psi \in \Phi$.

The auxiliary parentheses symbols are usually employed to disambiguate the reading of a sentence. The equivalence connective $\equiv$ is defined as an abbreviation, i.e., $\varphi \equiv \psi$ is an abbreviation of $(\varphi \supset \psi) \wedge (\psi \supset \varphi)$.

The semantic models of PL are simply truth assignments to the primitive propositions. Mathematically, an interpretation (or a model) of a PL is a function $\pi : \Phi_0 \rightarrow \{0, 1\}$. The interpretation π can be extended to the whole domain Φ by the following recursion rules:

1. $\pi(\neg\varphi) = 1 - \pi(\varphi)$,

2. $\pi(\varphi \wedge \psi) = \min(\pi(\varphi), \pi(\psi))$,

3. $\pi(\varphi \vee \psi) = \max(\pi(\varphi), \pi(\psi))$,

4. $\pi(\varphi \supset \psi) = \max(1 - \pi(\varphi), \pi(\psi))$.

We say that a wff φ is true (resp. false) under the interpretation π if $\pi(\varphi) = 1$ (resp. $\pi(\varphi) = 0$). A wff φ is valid if it is true under all interpretations and satisfiable if it is true under some interpretation. A wff that is not satisfiable is said to be unsatisfiable. Note that φ is valid iff $\neg\varphi$ is unsatisfiable.

2.2.2 Modal logic

As shown by the well-known Stone representation theorem, classical set theory has the intimate connection with the Boolean logic[83]. Analogously, rough set theory is closely related to modal logic[7]. The most well-known relationship is the connection of approximation space with possible world semantics for the modal epistemic logic $S5$.

The alphabet of $S5$ consists of a set of primitive propositions, Φ_0, and the logical symbols $\neg$(negation), $\wedge$(and), $\vee$(or), $\supset$(material implication), $\Box$(necessity modal operator), and $\Diamond$(possibility modal operator). The set of well-formed formulas (wffs) of $S5$ is defined as the smallest set Φ such that $\Phi_0 \subseteq \Phi$ and

- if $\varphi \in \Phi$, then $\neg\varphi, \Box\varphi$, and $\Diamond\varphi \in \Phi$

- if φ and $\psi \in \Phi$, then $\varphi \wedge \psi, \varphi \vee \psi$, and $\varphi \supset \psi \in \Phi$.

A Kripke model for $S5$ is a triple $M = (W, R, \pi)$, where W is a set of possible worlds, R is an equivalence relation on W, called an accessibility relation, and $\pi : \Phi_0 \rightarrow 2^W$ is a truth assignment that map a primitive propositions to the set of worlds in which it is evaluated to be true. The function π can be extended to all wffs recursively in the following way:

1. $\pi(\neg\varphi) = W - \pi(\varphi)$

2. $\pi(\varphi \wedge \psi) = \pi(\varphi) \cap \pi(\psi)$

3. $\pi(\varphi \vee \psi) = \pi(\varphi) \cup \pi(\psi)$

4. $\pi(\varphi \supset \psi) = \pi(\neg\varphi) \cup \pi(\psi)$

5. $\pi(\Box\varphi) = \{w \mid \forall u((w, u) \in R \Rightarrow u \in \pi(\varphi))\}$

6. $\pi(\Diamond\varphi) = \{w \mid \exists u((w, u) \in R \wedge u \in \pi(\varphi))\}$

For each model M and wff φ, $\pi(\varphi)$ is called the truth set of φ (in M).

Obviously, if $M = (W, R, \pi)$ is a Kripke model for $S5$, then (W, R) is an approximation space, and for each wff φ, $\pi(\varphi)$ is a subset of W and denote some concept in the approximation space, so we can consider its lower and upper approximations. A direct but interesting relationship between $S5$ and rough set theory is then established as follows:

$$\underline{R}\pi(\varphi) = \pi(\Box\varphi),$$

$$\overline{R}\pi(\varphi) = \pi\Diamond\varphi).$$

2.2.3 Arrow logic (AL)

In this section, we review the basic syntax and semantics of AL in order to lay the foundation for the development of arrow decision logic. AL is the basic modal logic of arrows [61, 87]. An arrow can represent a state transition in a program's execution, a morphism in category theory, an edge in a directed graph, etc. In AL, an arrow is an abstract entity; however, we can usually interpret it as a concrete relationship between two objects, which results in a pair-frame model [61, 87]. We now present the syntax and semantics of AL.

The basic alphabet of AL consists of a countable set of propositional symbols, the Boolean connectives $\neg$ and $\vee$, the modal constant δ, the unary modal operator $\otimes$, and the binary modal operator $\circ$. The set of AL wffs is the smallest set containing the propositional symbols and δ, closed under the Boolean connectives $\neg$ and $\vee$, and satisfying

- if φ is a wff, then $\otimes\varphi$ is a wff too;
- if φ and ψ are wffs, then $\varphi \circ \psi$ is also a wff.

In addition to the standard Boolean connectives, we also abbreviate $\neg \otimes \neg\varphi$ and $\neg(\neg\varphi \circ \neg\psi)$ as $\underline{\otimes}\varphi$ and $\varphi\underline{\circ}\psi$ respectively.

Semantically, these wffs are interpreted in arrow models.

Definition 2.5

1. *An arrow frame is a quadruple $\mathfrak{F} = (W, C, R, I)$ such that $C \subseteq W \times W \times W$, $R \subseteq W \times W$ and $I \subseteq W$.*

2. *An arrow model is a pair $\mathfrak{M} = (\mathfrak{F}, \pi)$, where $\mathfrak{F} = (W, C, R, I)$ is an arrow frame and π is a valuation that maps propositional symbols to subsets of W. An element in W is called an arrow in the model $\mathfrak{M}$.*

3. *The satisfaction of a wff φ on an arrow w of $\mathfrak{M}$, denoted by $w \models_{\mathfrak{M}} \varphi$ (as usual, the subscript $\mathfrak{M}$ can be omitted), is inductively defined as follows:*

   (a) $w \models p$ iff $w \in \pi(p)$ for any propositional symbol p,

   (b) $w \models \delta$ iff $w \in I$,

   (c) $w \models \neg\varphi$ iff $w \not\models \varphi$,

   (d) $w \models \varphi \vee \psi$ iff $w \models \varphi$ or $w \models \psi$,

(e) $w \models \varphi \circ \psi$ *iff there exist* s, t *such that* $(w, s, t) \in C$, $s \models \varphi$, *and* $t \models \psi$,

(f) $w \models \otimes\varphi$ *iff there is a* t *with* $(w, t) \in R$ *and* $t \models \varphi$.

Example 2.2 *Let us use the (multi-)graph shown in Figure 2.1 to explain the basic concept of arrow models. As shown in the figure, the arrow frame is characterized as* $\mathfrak{F} = (W, C, R, I)$, *where*

- $W = \{a_1, a_2, \ldots, a_{10}\}$,
- $C = \{(a_1, a_4, a_5), (a_1, a_4, a_7), \ldots, (a_8, a_8, a_{10})\}$
- $R = \{(a_5, a_6), \ldots, (a_{10}, a_{10})\}$
- $I = \{a_9, a_{10}\}$

If x_1, x_2, x_3, *and* x_4 *denote four cities and* $a_i (1 \leq i \leq 10)$ *denote routes between these them, then* I *denote the set of intra-city routes, whereas the others are inter-cities routes. A route* a_i *is a reverse route of another route* a_j *if* $(a_i, a_j) \in R$. *For example,* a_6 *is a reverse route of* a_5. *Also,* $(a_i, a_j, a_k) \in C$ *if* a_j *followed by* a_k *is an alternative route of* a_i. *For example,* a_3 *is a direct route connecting the cities* x_1 *and* x_3. *However, alternatively, we can also go from* x_1 *to* x_2 *through route* a_4 *and then from* x_2 *to* x_3 *through the route* a_5. *Thus,* a_4 *followed by* a_5 *is an alternative route to* a_3. *Now, let us consider an arrow logic language with two propositional symbols* p *and* q *meaning "the route is in congestion" and "the route is in bad situation" respectively. Assume that the valuation* π *of the arrow model is given as follows:*

$$\pi(p) = \{a_1, a_2, a_3, a_9, a_{10}\},$$

$$\pi(q) = \{a_1, a_4, a_6, a_8, a_{10}\}.$$

Then, in the model $(\mathfrak{F}, \pi)$, *we have* $a_9 \models \delta$ *since* a_9 *is an intra-city route. We also have* $a_7 \models \neg q \wedge (\otimes q)$ *which means that* a_7 *is not in bad situation, but one of its reverse routes is. Furthermore, we have* $a_3 \models p \wedge (q \circ (\neg p \wedge \neg q))$ *which means that* a_3 *is in congestion and there is an alternative route with a section* (a_4) *in bad situation followed by a section* (a_5) *neither in bad situation nor in congestion.*

Intuitively, in the arrow frame (W, C, R, I), W can be regarded as the set of edges of a directed graph; I denotes the set of identity arrows[3]; $(w, s) \in R$ if s is a reversed arrow of w; and $(w, s, t) \in C$ if w is an arrow composed of s and t. This intuition is reflected in the following definition of pair frames.

Definition 2.6 *An arrow frame* $\mathfrak{F} = (W, C, R, I)$ *is a pair frame if there exists a set* U *such that* $W \subseteq U \times U$ *and*

1. *for* $x, y \in U$, *if* $(x, y) \in I$ *then* $x = y$,
2. *for* $x_1, x_2, y_1, y_2 \in U$, *if* $((x_1, y_1), (x_2, y_2)) \in R$, *then* $x_1 = y_2$ *and* $y_1 = x_2$,
3. *for* $x_1, x_2, x_3, y_1, y_2, y_3 \in U$, *if* $((x_1, y_1), (x_2, y_2), (x_3, y_3)) \in C$, *then* $x_1 = x_2$, $y_2 = x_3$, *and* $y_1 = y_3$.

An arrow model $\mathfrak{M} = (\mathfrak{F}, \pi)$ *is called a pair model if* $\mathfrak{F}$ *is a pair frame. A pair model is called a (full) square model if the set of arrows* $W = U \times U$.

[3] An identity arrow is an arrow that has the same starting point and endpoint.

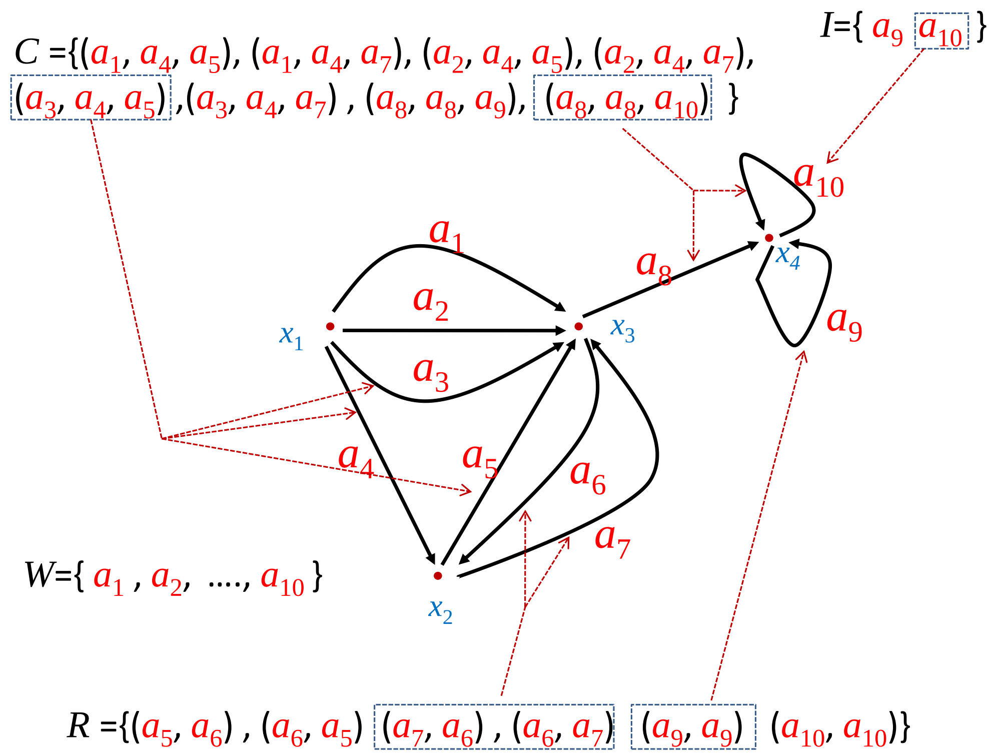


Figure 2.1: An arrow model

2.3 Knowledge Management

2.3.1 Knowledge management process

Although there is no commonly accepted definition of knowledge management, the following one mentioned in [72] is rather conceivable:

> *Knowledge management (KM) is a discipline that provides strategy, process, and technology to share and leverage information and expertise that will increase our level of understanding to more effectively solve problems and make decisions.*

According to this definition, it is believed that the objective of knowledge management systems is to support creation, transfer, and application of knowledge in organizations [3].

To uncover some assumptions about knowledge that underlie organizational KM process and KMS, several perspectives on knowledge are summarized in [3]. These include

1. the contrast of knowledge with data and information,
2. knowledge as a state of mind,
3. knowledge as an object,

4. knowledge as a process,
5. knowledge as a condition of having access to information, and
6. knowledge as a capability

For the contrast of knowledge with data and information, it is reiterated that data is raw numbers and facts, information is processed data, and knowledge is authenticated information. For knowledge as a state of mind, knowledge is described as "a state or fact of knowing" with knowing being a condition of "understanding gained through experience or study; the sum or range of what has been perceived, discovered, or learned". For knowledge as an object, it means that knowledge can be viewed as a thing to be stored and manipulated (i.e., an object). For knowledge as a process, it is emphasized that knowledge can be viewed as a process of simultaneously knowing and acting. For knowledge as a condition of having access to information, organizational knowledge must be organized to facilitate access to and retrieval of content. It is thought of as an extension of the view of knowledge as an object, with a special emphasis on the accessibility of the knowledge objects. For knowledge as a capability, knowledge can be viewed as a capability with the potential for influencing future action. However, it is also suggested that knowledge is not so much a capability for specific action, but the capacity to use information; learning and experience result in an ability to interpret information and to ascertain what information is necessary in decision making. (See [3] for further references).

These different perspectives of knowledge lead to different emphasis on how knowledge should be managed. Among them, the process view focuses on knowledge flow and the processes of creation, sharing, and distribution of knowledge. This is closely related to the KM life cycle—creation, capture, organization, and dissemination/sharing[72].

Any KM process starts from the creation of knowledge. The theory of organizational knowledge creation proposed in [66] is concerned with developing new content or replacing existing content within the organization's tacit and explicit knowledge. According to [3],

> *rooted in action, experience, and involvement in a specific context, the tacit dimension of knowledge (henceforth referred to as tacit knowledge) is comprised of both cognitive and technical elements. The cognitive element refers to an individual.s mental models consisting of mental maps, beliefs, paradigms, and viewpoints. The technical component consists of concrete know-how, crafts, and skills that apply to a specific context.*

On the other hand, the explicit knowledge is defined as follows [3]:

> *the explicit dimension of knowledge (henceforth referred to as explicit knowledge) is articulated, codified, and communicated in symbolic form and/or natural language.*

Based on such tacit-explicit knowledge classification, four modes of knowledge creation have been identified: socialization, externalization, internalization, and combination [66]. These modes of knowledge creation is explicated in [3] as follows:

> *The socialization mode refers to conversion of tacit knowledge to new tacit knowledge through social interactions and shared experience among organizational members*

(e.g., apprenticeship). The combination mode refers to the creation of new explicit knowledge by merging, categorizing, reclassifying, and synthesizing existing explicit knowledge (e.g., literature survey reports). The other two modes involve interactions and conversion between tacit and explicit knowledge. Externalization refers to converting tacit knowledge to new explicit knowledge (e.g., articulation of best practices or lessons learned). Internalization refers to creation of new tacit knowledge from explicit knowledge (e.g., the learning and understanding that results from reading or discussion).

However, as pointed out in [91], the term "tacit knowledge" should be replaced by the more appropriate "implicit knowledge":

> *Nonaka and Takeuchi put forward the proposition, embodied in the diagram, that "tacit knowledge" is somehow derived from explicit knowledge and, by other means, is made explicit. However, it is clear, from the analysis above, that implicit knowledge, which is not normally expressed, but may be expressed, is actually intended here. Implicit knowledge is that which we take for granted in our actions, and which may be shared by others through common experience or culture. For example, in establishing a production facility in a foreign country, a company knows it needs to acquire local knowledge of "how things are done here". Such knowledge may not be written down, but is known by people living and working in the culture and is capable of being written down, or otherwise conveyed to those who need to know. The knowledge is implicit in the way people behave towards one another, and towards authority, in that foreign culture, and the appropriate norms of behavior can be taught to the newcomers. Implicit knowledge, in other words, is expressible: tacit knowledge is not, and Nonaka would have saved a great deal of confusion had he chosen the more appropriate term.*

Apparently, the decision cases occurred in the past and stored in a data table may embed such a kind of implicit knowledge, and data mining is simply the IT tool to make implicit knowledge explicit. Thus, KDD is a process of knowledge creation from raw data. Once knowledge is created, its storage, organization, and retrieval, which are also referred as organizational memory, become crucial for effective organizational knowledge management. Organizational memory includes knowledge residing in various component forms, including written documentation, structured information stored in electronic databases, codified human knowledge stored in expert systems, and so on [3].

Several technologies to support the KM process are identified in [72]. These technologies include hybrid expert systems; personalization–profiling and customization; taxonomies, search, or knowledge discovery; knowledge metrics; and knowledge visualization. Among them, the core requirement of hybrid expert systems is "to capture the knowledge of experts and translate them into rules and reasoning processes to aid in decision support. Rules may range from simple and rigid to complex and vague" [72]. The objective of our work is to provide a class of logics that can represent such rules and reasoning processes.

The phase next to knowledge organization is knowledge transfer. Knowledge transfer depends on a learning process. Communication process and information flows drive knowledge transfer in organizations. To ease the communication process and reduce the cost of conversion between different forms of knowledge, a uniform style of knowledge representation formalism is expected.

Since the logics proposed in this thesis are all the same style—they are all based on extensions of the decision logic, we can achieve the purpose of easy communication between different units or organizations.

The last phase of knowledge management is the application of knowledge. It is emphasized in [3] that the source of competitive advantage of KM resides in the application of the knowledge rather than in the knowledge itself. There are three primary mechanisms for the integration of knowledge to create organizational capability: directives, organizational routines, and self-contained task teams[3]. An important problem of knowledge application may be "deciding upon the rules and routines to apply to a problem, given that over time, the organization has learned and codified a large number of rules and routines, so that choosing which rules to activate for a specific choice making scenario is itself problematic"[3]. The technology that can help in this phase is the reasoning power of decision-support systems. As mentioned in Chapter 1, our logic-based representation facilitates the easy integration of the knowledge bases with inference engines of the decision-support systems.

The knowledge management process is summarized in Figure 2.2, where knowledge representation is highlighted with red color to reiterate the main theme of our work.

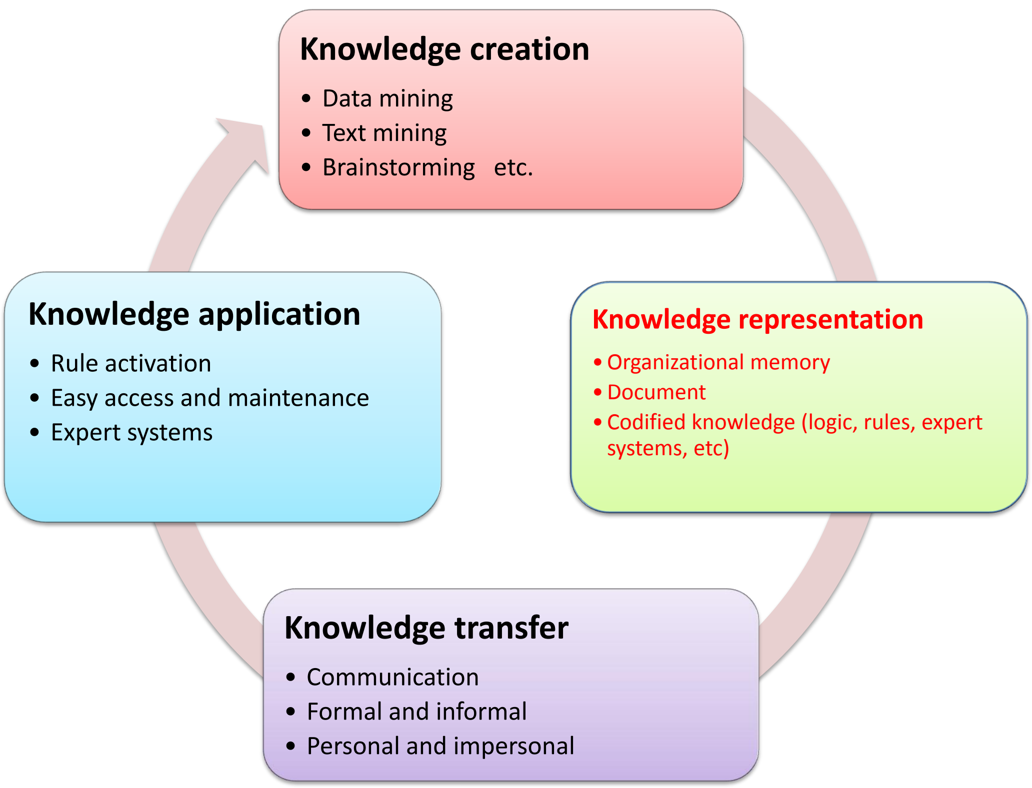


Figure 2.2: The knowledge management life cycle

2.3.2 Logic-based knowledge management

In [22], a logic-based approach to KM is advocated for meeting the requirement of system integration. The requirement occurs when an organization have a large asset of heterogenous computer programs, databases and applications. It will be ideal to integrate all these systems into a single homogenous system. In real life, however, this would be an unrealistic task because it requires a lot of efforts and a long development time. Therefore, a more realistic approach to system integration is to support the use of the existing systems in a way most comfortable for the end-user, hide the technical difficulties caused by the heterogenous databases and/or applications, and provide tools for combining information coming from various sources[22]. In order to find an intelligent solution for the KM oriented support of system integration, the most important thing is to choose a synthetic and uniform technology basis.

Logic-based paradigm is chosen in [22] because of its important advantages in the following aspects:

- declarative description of meta-information on information sources,
- symbolic manipulation for generating database queries,
- openness, knowledge-based customizability,
- deduction in intelligent answering,
- learning capacity wherever it is possible,
- natural language interface.

The paradigm results in the development of the SILK project (System Integration via Logic in Knowledge Management). The objective of SILK is to show the applicability of the logic-based knowledge management approach to system integration. Although the logic employed in the SILK project is different than the DL-styled logics used here, our approach also inherits the advantages of logic-based KM. In particular, we can have a uniform style of knowledge representation formalism. This will make the organization and application of knowledge more accessible for the enterprize. We will use a running scenario to illustrate how effective human resource management (HRM) can be achieved via such a representation.

2.3.3 A running scenario

The recruitment and selection of high-potential talent are important for the enterprize, since the human capital is the key to success in a knowledge-based economy. However, it is difficult in the selection stage to predict the work performance of the applicants. Conventional selection approaches including basic competency test, professional skill test, project proposal test, and interviews have been widely used for many years. Data mining methodology to assist the decision makers in identifying the most suitable talents has been previously proposed in [10]. The approach is developed to explore human resource data and thus derive decision rules between personnel characteristics and work behaviors. Let us now consider a high-tech company (say, the Knowledge Technology Corporation, abbreviated as KTC), which is recruiting talent to fill different positions including program designers, system analysts, project managers, and researcher.

Example 2.3 To warm up, let us consider how DL can be used to represented decision rules for the preliminary classification of the applicants to KTC recruitment. To achieve this task, we retrieve the basic data of well-performed employees from the personnel data base of the company. The definition of well-performed employees depends on the requirement of the positions. For example, an employee may be considered as well-performed if the evaluation of his/her performance in the last five years is above some level. It is believed that these employees fit their current positions quite well. We assume the basic data of employees contains the following attributes: age (when hired), gender, educational background (degree), and major subject, and current position. The last attribute is the decision attribute and the others are condition attributes. For the sake of simplicity, we will denote attributes by lower case italic letters. Thus, the attributes and their domains are coded as follows:

1. i_1: the age of an employee when he/she is hired
   - 1: 25 years old and below
   - 2: 26 to 30 years old
   - 3: 31 to 35 years old
   - 4: 36 years old and above
2. i_2: the gender of an employee
   - F: female
   - M: male
3. i_3: education background (the highest degree that an employee possesses)
   - JC: junior college degree and below
   - BA: bachelor degree
   - MS: master degree and above
4. i_4: the major of an employee
   - EE: electrical engineering
   - CS: computer science
   - MH: mathematics
   - IM: information management
5. d: the current position of an employee
   - PD: program designers
   - SA: system analysts
   - PM: project managers
   - RD: researcher

The retrieved employee data is then taken as our input decision table and appropriate data mining algorithm is applied to derive decision rules. Since the main focus of the work is on the representation of the rules and the decision table just looks like that appears in Example 2.1, we will not write down the decision table again. Also, it is assumed that pre-determined thresholds for confidences and supports are used to select the decision rules, so we come up with a set of decision rules in the form of DL rules. For example, we may have a rule like this

$$(i_3, \mathrm{MS}) \wedge ((i_4, , \mathrm{MH}) \vee (i_4, , \mathrm{CS})) \longrightarrow (d, \mathrm{RD}).$$

The set of all derived decision rules (perhaps with the background knowledge) constitutes the first knowledge base KB_1 for the recruitment process. Now, given the basic data of the applicants, each applicant can be classified as appropriate for one (or more) particular position(s). An applicant classified as appropriate for a particular position is consider as a candidate of that position. For example, an applicant whose major is mathematics and highest degree is master is consider as a candidate for the researcher position according to the above rule. Thus, for each position, we have a set of candidates for further process. The process of the preliminary classification phase is shown in Figure 2.3, where the knowledge base is colored red to emphasize the role of DL-styled logics.

2.4 Related Work

When rough set theory is applied to multi-criteria decision analysis (MCDA), it is crucial that preference-ordered attribute domains and decision classes be dealt with [29, 30, 31, 33, 34, 35, 36, 82]. Thus, the indiscernibility relation in rough set theory is replaced by a dominance relation to solve the multi-criteria sorting problem, and the data table is replaced by a pairwise comparison table to solve multi-criteria choice and ranking problems. The approach is called the dominance-based rough set approach (DRSA).

A strong assumption about data tables is that each object takes exactly one value with respect to an attribute. However, in practice, we may only have incomplete information about the values of an object's attributes. Thus, more general data tables and decision logics are needed to represent and reason about incomplete information. For example, set-valued and interval set-valued data tables have been introduced to represent incomplete information [51, 52, 53, 60, 94]. DRSA has also been extended to deal with missing values in MCDA problems [34, 82].

The notion of IS-morphism introduced in Chapter 4 is related to the work in [39]. The algebraic properties of IS-morphism between functional information systems (FIS) was previously studied there under the name of O-A-D homomorphism[4]. Our notion of IS-morphism between relational information systems (RIS) is a straightforward generalization of that between FIS. In fact, if FIS and RIS are considered as many-sorted algebras [6], both IS-morphism and O-A-D homomorphism can be seen as homomorphism in universal algebra [8, 11].

The investigation of RIS also facilitates a further generalization of rough set theory. In classical rough set theory, lower and upper approximations are defined in terms of indiscernibility relations based on functional information associated with the objects. However, it has been noted that many applications, such as social network analysis [74], need to represent both functional and

[4] O, A, and D denotes objects, attributes, and the domain of values respectively.

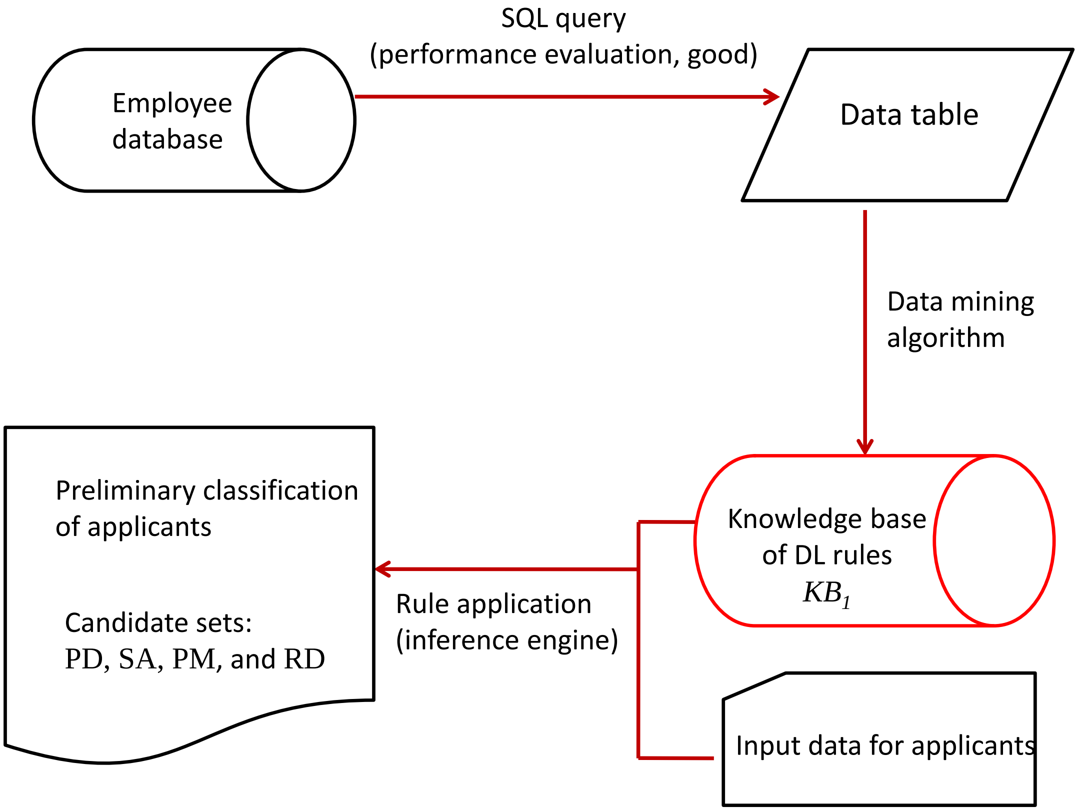


Figure 2.3: The preliminary classification phase of the recruitment process

relational information. Based on this observation, a concept of relational granulation was recently proposed in [57].

Chapter 3

Decision Logics for Multicriteria Decision Analysis

When rough set theory is applied to multi-criteria decision analysis (MCDA), it is crucial that preference-ordered attribute domains and decision classes be dealt with [29, 30, 31, 33, 34, 35, 36, 82]. The original rough set theory cannot handle inconsistencies arising from violations of the dominance principle due to its use of the indiscernibility relation. Therefore, in the above-mentioned work, the indiscernibility relation is replaced by a dominance relation to solve the multi-criteria sorting problem, and the data table is replaced by a pairwise comparison table to solve multi-criteria choice and ranking problems. The approach is called the dominance-based rough set approach (DRSA). For MCDA problems, DRSA can induce a set of decision rules from sample decisions provided by decision-makers. The induced rules form a comprehensive preference model and can provide recommendations about a new decision-making environment.

A strong assumption about data tables is that each object takes exactly one value with respect to an attribute. However, in practice, we may only have incomplete information about the values of an object's attributes. Thus, more general data tables and decision logics are needed to represent and reason about incomplete information. For example, set-valued and interval set-valued data tables have been introduced to represent incomplete information [51, 52, 53, 60, 94]. A generalized decision logic based on interval set-valued data tables is also proposed in [94]. In these formalisms, the attribute values of an object may be a subset or an interval set in the domain. Since crisp subsets and interval sets are both special cases of fuzzy sets, further generalization of data tables is desirable to represent uncertain information. In data tables containing such information, an object can take a fuzzy subset of values for each attribute. To represent knowledge induced from uncertain data tables, the decision logic also needs to be generalized.

DRSA has also been extended to deal with missing values in MCDA problems [34, 82]. A data table with missing values is a special case of uncertain data tables. Therefore, we propose further extending DRSA to uncertain data tables and fuzzy data tables. In this chapter, we present a logical treatment of DRSA in precise data tables, as well as uncertain and fuzzy data tables. Our approach is concerned with variants of DL for data tables.

The remainder of the chapter is organized as follows. In Sections 3.1 to 3.4, we respectively present generalized DL for preference-ordered data tables, preference-ordered uncertain data tables, preference-ordered fuzzy data tables, and pairwise comparison tables. For each logic, the syntax and semantics are described, and some quantitative measures for the rules of the logics are defined. Finally, in Section 3.5, we discuss the main contribution of this chapter and indicate the direction of future research.

3.1 Preference-ordered Data Tables

For MCDA problems, each object in a data table or decision table can be seen as a sample decision, and each condition attribute is a criterion for the decision. Since the domain of values of a criterion is usually ordered according to the decision-maker's preferences, we define a preference-ordered data table (PODT) as a tuple

$$T = (U, A, \{(V_i, \succeq_i) \mid i \in A\}, \{f_i \mid i \in A\}),$$

where $T = (U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$ is a classical data table; and for each $i \in A$, $\succeq_i \subseteq V_i \times V_i$ is a binary relation over V_i. The relation $\succeq_i$ is called a *weak preference relation* or *outranking* on V_i, and represents a preference over the set of objects with respect to the criterion i [82]. For $x, y \in U$, $f_i(x) \succeq_i f_i(y)$ means "x is at least as good as y with respect to criterion i.

To represent the rules induced from a PODT, we introduce preference-ordered decision logic (PODL). The syntax of PODL is the same as that of DL, except for the form of the atomic formulas. An atomic formula in PODL is a descriptor in the form of $(\geq_i, v)$ or $(\leq_i, v)$, where $i \in A$ and $v \in V_i$. The satisfaction relation between U and the set of PODL wffs is defined in the same way as for DL wffs, except that the satisfaction of an atomic formula is defined by $x \models (\geq_i, v)$ iff $f_i(x) \succeq v$, and by $x \models (\leq_i, v)$ iff $v \succeq f_i(x)$. Other semantic notions in DL, such as validity, support, and confidence, can all be used in the case of PODL without any modifications. The confidence function for PODL rules has also been defined by [37].

In [34], three types of rules are explicitly identified. We translate these rules into PODL rules as follows.

1. $\bigwedge_{i \in C}(\geq_i, v_i) \longrightarrow (\geq_d, v_d)$, where $C \subseteq A$ is a subset of condition attributes, $d \in A \backslash C$ is a decision attribute, $v_i \in V_i$ for all $i \in C$, and $v_d \in V_d$.

2. $\bigwedge_{i \in C}(\leq_i, v_i) \longrightarrow (\leq_d, v_d)$, where $C \subseteq A$ is a subset of condition attributes, $d \in A \backslash C$ is a decision attribute, $v_i \in V_i$ for all $i \in C$, and $v_d \in V_d$.

3. $(\bigwedge_{i \in C_1}(\geq_i, v_i) \wedge \bigwedge_{i \in C_2}(\leq_i, v_i)) \longrightarrow ((\geq_d, v_d) \wedge (\leq_d, v'_d))$, where $C_1 \cup C_2 \subseteq A$ is a subset of condition attributes, $d \in A \backslash (C_1 \cup C_2)$ is a decision attribute, $v_i \in V_i$ for all $i \in C_1 \cup C_2$, and $v_d, v'_d \in V_d$.

Example 3.1 Continuing with Example 2.1, let us assume that each $V_i (i = o, p, t, d)$ is now endowed with a weak preference relation $\succeq_i$ such that $4 \succeq_i 3 \succeq_i 2 \succeq_i 1$. Thus, we have atomic formulas like $(\geq_o, 4)$, $(\geq_p, 1)$, and $(\geq_t, 2)$. Let us now consider the following rules:

$$r_1 = (\geq_o, 3) \longrightarrow (\geq_d, 3),$$

$$r_2 = (\leq_p, 2) \longrightarrow (\leq_d, 2),$$

$$r_3 = (\geq_o, 4) \wedge (\leq_t, 2) \longrightarrow (\geq_d, 2) \wedge (\leq_d, 3).$$

Then, we have

	asp	rsp	cfd
r_1	5	$\frac{1}{2}$	$\frac{5}{8}$
r_2	5	$\frac{1}{2}$	$\frac{5}{6}$
r_3	2	$\frac{1}{5}$	1

Among these rules, only r_3 is valid. ■

3.1.1 The running scenario

Let us now continue the recruitment process of the KTC and conduct the professional skills test for the applicants. It is assumed that the professional skills of an employee are related to his/her short-term performance since the required skills of high-tech industry are rapidly changing. Note that in the preliminary classification stage, we have classified each applicant into appropriate position class(es). The professional skills required by different positions may be different. However, for simplification, we assume that all applicants are tested with the same set of professional skills, which includes programming languages, engineering mathematics, and software engineering. The test result of each applicant is given in a 10-point score. To utilize the past experience, we retrieve the professional skill test results of the employees hired in the last three years and their performance evaluation. The performance evaluation data of the employees may be stored in a separated database. We assume the company has established a performance management system to evaluate employees performance. Based on the performance, the employees will be ranked into four categories: outstanding (A), good (B), fair (C), and poor (D). By joining the data retrieved from the databases of professional skill test results and employees' performance evaluation, we obtain a PODT with three decision attributes and one condition attribute. The attributes and their domains are coded as follows:

1. condition attributes:
   - p: programming languages, $V_p = \{0, 1, 2, \ldots, 10\}$;
   - e: engineering mathematics, $V_e = \{0, 1, 2, \ldots, 10\}$;
   - s: software engineering, $V_s = \{0, 1, 2, \ldots, 10\}$;
2. decision attribute:
   - d: performance evaluation, $V_d = \{A, B, C, D\}$

It is assumed that V_p, V_e, and V_s are all endowed with the weak preference ordering $10 \succeq_i 9 \succeq_i \cdots \succeq_i 0$ and V_d is endowed with the ordering $A \succeq_d B \succeq_d C \succeq_d D$. As in the case of Example 2.3, we assume an appropriate data mining algorithm (e.g., the DRSA approach) is applied to derive PODL rules. Also, it is still assumed that pre-determined thresholds for confidences and supports are used to select the decision rules, so finally, we have a set of PODL rules that constitute the second knowledge base KB_2 of the recruitment process. A typical rule in KB_2 is like this

$$(\geq_p, 6) \wedge (\geq_e, 8) \wedge (\geq_s, 6) \longrightarrow (\geq_d, B).$$

Now, given the professional skills test result of an applicant, we can apply the PODL rules to predict his/her short-term performance in the following way. Because there may be more than one rules in KB_2 whose antecedents can match the applicant's test result, we will apply all applicable rules to the applicant. Let x be an applicant and $\varphi \longrightarrow \psi$ is a PODL rule in KB_2, then we will write $x \models \psi$ if the test result of x matches with φ. For example, if the test result of x is programming languages, 8, engineering mathematics, 9 and software engineering, 7, then we can write $x \models (\geq_d, B)$. Considering all rules applicable to x, we denote $sp_l(x) = \max\{v \mid x \models ((\geq_d, v)\}$ and $sp_u(x) = \min\{v \mid x \models ((\leq_d, v)\}$. Intuitively, $sp_l(x)$ and $sp_u(x)$ denote the lower and upper bounds of x's short-term performance prediction respectively. We can further define the set of grades $sp(x) = \{v \in V_d \mid sp_l(x) \leq v \leq sp_u(x)\}$ for each applicant x. Thus, after the short-term performance prediction phase, each applicant is associated with a set of grades denoting the prediction on his/her short-term performance. Note that, in reality, inconsistency may exist in the original PODT (i.e., the dominance principle may be violate). In the presence of inconsistency, the derived rules may be also mutually inconsistent and $sp(x)$ may be empty for some x. If it is the case, we will set $sp(x)$ to be V_d. The process of this phase is shown in Figure 3.1.

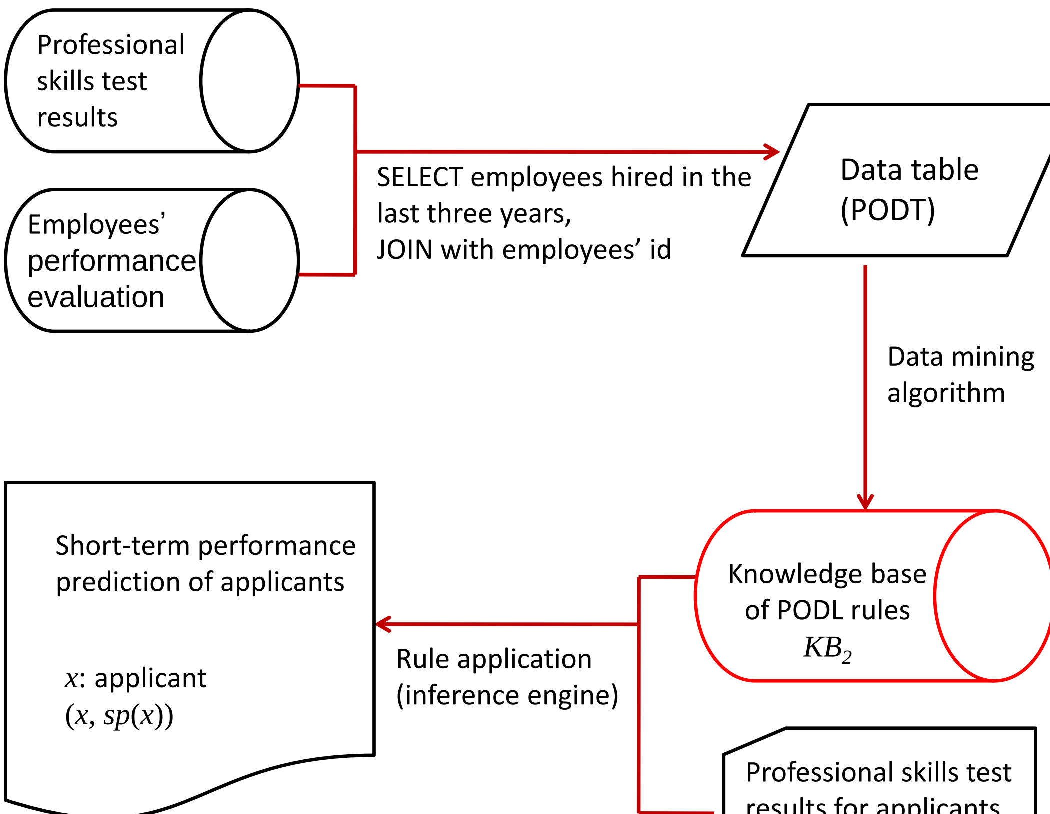


Figure 3.1: The short-term performance prediction phase of the recruitment process

3.2 Preference-ordered Uncertain Data Tables

PODL is suitable for the representation of rules induced from a PODT. However, the latter inherits the restriction of classical DT so that uncertain information can not be represented. An uncertain data table is a generalization of DT such that the values of some or all attributes are imprecise [16, 13]. An analogous generalization can be applied to PODT to define preference-ordered uncertain data tables (POUDT). Formally, a POUDT is a tuple

$$T = (U, A, \{(V_i, \succeq_i) \mid i \in A\}, \{f_i \mid i \in A\}),$$

where $U, A, \{(V_i, \succeq_i) \mid i \in A\}$ are defined as above, and for each $i \in A$, $f_i : U \rightarrow 2^{V_i} - \{\emptyset\}$. The intuition about POUDT is that the value of attribute i of an object x belongs to $f_i(x)$, though the value is not known exactly. When $f_i(x)$ is a singleton, we say that the value is precise. If all attribute values of T are precise, then T is said to be single-valued.

PODL is also generalized to preference-ordered uncertain decision logic (POUDL). The syntax of POUDL is same as that of PODL, except that its atomic formulas are of the form (i, s_i), where $i \in A$ and $s_i \subseteq V_i$. When $s_i = \{v \in V_i \mid v \succeq_i v_i\}$ (resp. $s_i = \{v \in V_i \mid v_i \succeq_i v\}$), we abbreviate (i, s_i) as $(\geq_i, v_i)$ (resp. $(\leq_i, v_i)$). To define the semantics of POUDL, we must first rewrite each wff into its normal form.

A wff is in a *conjunctive normal form* (CNF) if it is a conjunction of formulas of the form $\vee_{i \in B}(i, s_i)$, where $B \subseteq A$ is a subset of mutually distinct attributes. A wff is in a *disjunctive normal form* (DNF) if it is a disjunction of formulas of the form $\wedge_{i \in B}(i, s_i)$, where $B \subseteq A$ is a subset of mutually distinct attributes. Given a POUDL wff φ, its CNF and DNF are denoted by φ^c and φ^d respectively. Any POUDL wff can be rewritten in both CNF and DNF by using Boolean algebra and the following rewriting rules:

$$\neg(i, s) = (i, V_i \backslash s)$$

$$(i, s_1) \vee (i, s_2) = (i, s_1 \cup s_2)$$

$$(i, s_1) \wedge (i, s_2) = (i, s_1 \cap s_2).$$

For the semantics of POUDL, we define the positive satisfaction relation $\models^+$ for CNF formulas and negative satisfaction relation $\models^-$ for DNF formulas. The definition is as follows:

1. $x \models^+ (i, s)$ iff $f_i(x) \subseteq s$,
2. $x \models^+ \varphi \vee \psi$ iff $x \models^+ \varphi$ or $x \models^+ \psi$,
3. $x \models^+ \varphi \wedge \psi$ iff $x \models^+ \varphi$ and $x \models^+ \psi$,
4. $x \models^- (i, s)$ iff $f_i(x) \cap s = \emptyset$,
5. $x \models^- \varphi \wedge \psi$ iff $x \models^- \varphi$ or $x \models^- \psi$,
6. $x \models^- \varphi \vee \psi$ iff $x \models^- \varphi$ and $x \models^- \psi$.

Then, for any POUDL wff φ, we define $x \models^+ \varphi$ iff $x \models^+ \varphi^c$, and $x \models^- \varphi$ iff $x \models^- \varphi^d$. According to the semantics of POUDL, $x \models^+ (\geq_i v_i)$ if for all $v \in f_i(x)$, v is preferred over v_i with respect to the criterion i. Therefore, we can be sure that, if $x \models^+ (\geq_i v_i)$ holds, then the value of criterion i of x will at least reach the level of v_i no matter what the actual value is. Analogously, if $x \models^- (\geq_i v_i)$ holds, we can be sure that the value of criterion i of x will not be above the level of v_i no matter what the actual value is.

For each POUDL wff φ and a given POUDT T, we define two meaning sets:

$$m_T^+(\varphi) = \{x \in U \mid x \models^+ \varphi\},$$

$$m_T^-(\varphi) = \{x \in U \mid x \models^- \varphi\};$$

$m_T^+(\varphi)$ is the set of objects that are known to satisfy φ, and $m_T^-(\varphi)$ is the set of objects that are known not to satisfy φ. The indeterminate region of φ with respect to T is defined as

$$m_T^*(\varphi) = U \backslash (m^+(\varphi) \cup m_T^-(\varphi)).$$

As usual, the subscript T can be omitted if it is clear from the context. Using the notations from rough set theory, we also define

$$\underline{m}(\varphi) = m^+(\varphi) \text{ and } \overline{m}(\varphi) = U \backslash m^-(\varphi).$$

Note that the three types of rules mentioned in Section 3.1 can also be represented in POUDL, though the semantics is quite different.

The quantitative measures of the rules' usefulness can be defined by the notion of completion of a POUDT. Let $T = (U, A, \{(V_i, \succeq_i) \mid i \in A\}, \{f_i \mid i \in A\})$ be a POUDT. Then, a PODT $S = (U, A, \{(V_i, \succeq_i) \mid i \in A\}, \{f_i' \mid i \in A\})$ is a *completion* of T if $f_i'(x) \in f_i(x)$ for all $i \in A$ and $x \in U$. The number of completions of T is equal to $\prod_{i \in A, x \in U} |f_i(x)|$. Let $CL(T)$ denote the set of all completions of T. If we identify a singleton set with its element by slightly abusing the notation, then a completion of a POUDT (i.e., a PODT) can be considered as a special case of POUDT. This yields the following definition.

Definition 3.1 *Let $T = (U, A, \{(V_i, \succeq_i) \mid i \in A\}, \{f_i \mid i \in A\})$ be a POUDT and $\varphi \longrightarrow \psi$ be a POUDL rule, then*

1. *$\varphi \longrightarrow \psi$ is strongly valid in T if $\overline{m}(\varphi) \subseteq m^+(\psi)$ and weakly valid in T if $m^+(\varphi) \subseteq m^+(\psi)$;*

2. *the absolute support interval of $\varphi \longrightarrow \psi$ is*

$$asi_T(\varphi \longrightarrow \psi) = [\min_{S \in CL(T)} asp_S(\varphi \longrightarrow \psi), \max_{S \in CL(T)} asp_S(\varphi \longrightarrow \psi)];$$

3. *the relative support interval of $\varphi \longrightarrow \psi$ is*

$$rsi_T(\varphi \longrightarrow \psi) = [\min_{S \in CL(T)} rsp_S(\varphi \longrightarrow \psi), \max_{S \in CL(T)} rsp_S(\varphi \longrightarrow \psi)]; \text{ and}$$

4. *the confidence interval of $\varphi \longrightarrow \psi$ is*

$$cfi_T(\varphi \longrightarrow \psi) = [\min_{S \in CL(T)} cfd_S(\varphi \longrightarrow \psi), \max_{S \in CL(T)} cfd_S(\varphi \longrightarrow \psi)].$$

The next two propositions show how these measures are calculated.

Proposition 3.1 *Let φ be a POUDL wff and T be a POUDT. Then, for all $x \in Uni(T)$, we have*

1. $x \models_T^+ \varphi$ *iff* $x \models_S \varphi$ *for all* $S \in CL(T)$,

2. $x \models_T^- \varphi$ *iff* $x \not\models_S \varphi$ *for all* $S \in CL(T)$, *and*

3. $x \in m_T^*(\varphi)$ *iff there exist* $S_1, S_2 \in CL(T)$ *such that* $x \models_{S_1} \varphi$ *and* $x \not\models_{S_2} \varphi$.

Proof: We first note that if S is a PODT, then for any POUDL wff φ, $x \models_S \varphi$ iff $x \models_S \varphi^c$ iff $x \models_S \varphi^d$. Thus, without loss of generality, we only need to consider wffs in CNF or DNF. Let us now prove the first equivalence. The second equivalence can be proved analogously, and the third follows from the first two.

($\Rightarrow$): If φ is in CNF, then we have $x \models_T^+ \varphi$ iff $x \models_T^+ \vee_{i \in B}(i, s_i)$ for each conjunct $\vee_{i \in B}(i, s_i)$ of φ. Now, $x \models_T^+ \vee_{i \in B}(i, s_i)$ implies that there exists $i \in B$ such that $f_i(x) \subseteq s_i$. This, in turn, implies that $x \models_S \vee_{i \in B}(i, s_i)$ for any $S \in CL(T)$. Thus, $x \models_T^+ \varphi$ implies that $x \models_S \varphi$ for all $S \in CL(T)$.

($\Leftarrow$): If φ is in CNF and $x \models_S \varphi$ for all $S \in CL(T)$, then for any conjunct $\vee_{i \in B}(i, s_i)$ of φ, we have $x \models_S \vee_{i \in B}(i, s_i)$ for any $S \in CL(T)$. Assume $x \not\models_T^+ \vee_{i \in B}(i, s_i)$ for some conjunct $\vee_{i \in B}(i, s_i)$ of φ; then $f_i(x) \not\subseteq s_i$ holds for all $i \in B$. Thus, since the attributes in B are mutually distinct, we can have an $S = (U, A, \{(V_i, \succeq_i) \mid i \in A\}, \{f_i' \mid i \in A\}) \in CL(T)$ such that $f_i'(x) \in f_i(x) \backslash s_i$ for all $i \in B$. Obviously, this implies that $x \not\models_S \vee_{i \in B}(i, s_i)$ and contradicts the fact that $x \models_S \vee_{i \in B}(i, s_i)$ for any $S \in CL(T)$. Therefore, we can derive $x \models_T^+ \vee_{i \in B}(i, s_i)$ for any conjunct $\vee_{i \in B}(i, s_i)$ of φ, and consequently, $x \models_T^+ \varphi$. ■

Proposition 3.2 *Let $\varphi \longrightarrow \psi$ be a POUDL rule and T be a POUDT, then we have*

1.

$$|\underline{m}(\varphi \wedge \psi)| = \min_{S \in CL(T)} asp_S(\varphi \longrightarrow \psi),$$

$$|\overline{m}(\varphi \wedge \psi)| = \max_{S \in CL(T)} asp_S(\varphi \longrightarrow \psi);$$

2.

$$\frac{|\underline{m}(\varphi \wedge \psi)|}{|U|} = \min_{S \in CL(T)} rsp_S(\varphi \longrightarrow \psi),$$

$$\frac{|\overline{m}(\varphi \wedge \psi)|}{|U|} = \max_{S \in CL(T)} rsp_S(\varphi \longrightarrow \psi);$$

3.

$$\frac{|\underline{m}(\varphi \wedge \psi)|}{|\overline{m}(\varphi) \backslash (m^*(\varphi) \cap m^-(\varphi \wedge \neg\psi))|} = \min_{S \in CL(T)} cfd_S(\varphi \longrightarrow \psi),$$

$$\frac{|\overline{m}(\varphi \wedge \psi)|}{|\overline{m}(\varphi) \backslash (m^*(\varphi) \cap m^-(\varphi \wedge \psi))|} = \max_{S \in CL(T)} cfd_S(\varphi \longrightarrow \psi).$$

Proof: The only non-trivial case to be proved is the third one. We first note that

$$\min_{S\in CL(T)} cfd_S(\varphi \longrightarrow \psi) = \min_{S\in CL(T)} \frac{|m_S(\varphi\wedge\psi)|}{|m_S(\varphi)|}.$$

Let S^* be a completion of T such that

$$S^* \in \arg\min_{S\in CL(T)} \frac{|m_S(\varphi\wedge\psi)|}{|m_S(\varphi)|}.$$

Then, for any object x, we can consider the following cases:

Case 1: If $x \in \overline{m}(\varphi)\backslash(m^*(\varphi)\cap m^-(\varphi\wedge\neg\psi))$ and $x \notin m_{S^*}(\varphi)$, then $x \notin \underline{m}(\varphi)$ since $\underline{m}(\varphi) \subseteq m_S(\varphi)$ for any $S \in CL(T)$. This means $x \in m^*(\varphi)$ and $x \notin m^-(\varphi\wedge\neg\psi))$, so there exists $S \in CL(T)$ such that $x \in m_S(\varphi\wedge\neg\psi))$ by Proposition 3.1. Since the attribute values of different objects in a completion can be independently determined, we can define $S' \in CL(T)$ such that for each attribute i, $f_i^{S'}(x) = f_i^S(x)$ and $f_i^{S'}(y) = f_i^{S^*}(y)$ for all $y \neq x$, where $f_i^{S'}, f_i^S$, and $f_i^{S^*}$ correspond to the attribute functions of S', S, and S^* respectively. Then $|m_{S'}(\varphi)| = |m_{S^*}(\varphi)| + 1$ and $|m_{S'}(\varphi\wedge\psi)| = |m_{S^*}(\varphi\wedge\psi)|$. This contradicts the minimality assumption for S^*.

Case 2: If $x \in m_{S^*}(\varphi)$ and $x \notin \overline{m}(\varphi)\backslash(m^*(\varphi)\cap m^-(\varphi\wedge\neg\psi))$, then $x \in m^*(\varphi)$ and $x \in m^-(\varphi\wedge\neg\psi)$, since $m_S(\varphi) \subseteq \overline{m}(\varphi)$ for any $S \in CL(T)$. From $x \in m_{S^*}(\varphi)$ and $x \in m^-(\varphi\wedge\neg\psi)$, we can derive $x \in m_{S^*}(\varphi\wedge\psi)$ by Proposition 3.1. From $x \in m^*(\varphi)$, we can find an $S \in CL(T)$ such that $x \not\models_S \varphi$ (and, of course, $x \not\models_S \varphi\wedge\psi$). Thus, we can also define $S' \in CL(T)$ such that for each attribute i, $f_i^{S'}(x) = f_i^S(x)$ and $f_i^{S'}(y) = f_i^{S^*}(y)$ for all $y \neq x$. Then, $|m_{S'}(\varphi)| = |m_{S^*}(\varphi)| - 1$ and $|m_{S'}(\varphi\wedge\psi)| = |m_{S^*}(\varphi\wedge\psi)| - 1$, which implies that

$$\frac{|m_{S'}(\varphi\wedge\psi)|}{|m_{S'}(\varphi)|} \leq \frac{|m_{S^*}(\varphi\wedge\psi)|}{|m_{S^*}(\varphi)|}.$$

Therefore, without loss of generality, we can assume that $m_{S^*}(\varphi) = \overline{m}(\varphi)\backslash(m^*(\varphi)\cap m^-(\varphi\wedge\neg\psi))$.

Now, if there exists $x \in m_{S^*}(\varphi\wedge\psi)$ and $x \notin \underline{m}(\varphi\wedge\psi)$, then $x \in \overline{m}(\varphi)\backslash(m^*(\varphi)\cap m^-(\varphi\wedge\neg\psi))$, so we can consider two possibilities.

Case 1: $x \in m^+(\varphi)$. Then, from the assumption, we can derive $x \in m^*(\psi)$

Case 2: $x \in m^*(\varphi)$ and $x \in m^*(\varphi\wedge\neg\psi)$.

In both cases, we can find an $S \in CL(T)$ such that $x \models_S \varphi\wedge\neg\psi$. Let S' be a completion of T that has the same attribute values as S for x and the same attribute values as S^* for objects that are not x. Then $|m_{S'}(\varphi)| = |m_{S^*}(\varphi)|$ and $|m_{S'}(\varphi\wedge\psi)| = |m_{S^*}(\varphi\wedge\psi)| - 1$. This contradicts the minimality assumption for S^*. Therefore, $m_{S^*}(\varphi\wedge\psi) = \underline{m}(\varphi\wedge\psi)$, which means that there exists an

$$S^* \in \arg\min_{S\in CL(T)} \frac{|m_S(\varphi\wedge\psi)|}{|m_S(\varphi)|},$$

such that

$$\frac{|\underline{m}(\varphi\wedge\psi)|}{|\overline{m}(\varphi)\backslash(m^*(\varphi)\cap m^-(\varphi\wedge\neg\psi))|} = cfd_{S^*}(\varphi \longrightarrow \psi).$$

Therefore, the result for the lower bound is proved. The result for the upper bound can be proved in an analogous way. ■

In general, the attribute symbols in the antecedent of a rule and those in the consequent are disjoint. For this special type of rule, the confidence interval can be simplified slightly.

Corollary 3.1 *Let A_φ denote the set of attributes appearing in a wff φ, and $\varphi \longrightarrow \psi$ be a POUDL rule such that $A_\varphi \cap A_\psi = \emptyset$. Then, $cfi(\varphi \longrightarrow \psi) = [low, up]$, where*

$$low = \frac{|\underline{m}(\varphi \wedge \psi)|}{|\overline{m}(\varphi)\backslash(m^*(\varphi) \cap m^+(\psi))|}, \text{ and}$$

$$up = \frac{|\overline{m}(\varphi \wedge \psi)|}{|\overline{m}(\varphi)\backslash(m^*(\varphi) \cap m^-(\psi))|}.$$

Proof: It suffices to show that $m^*(\varphi) \cap m^-(\varphi \wedge \neg\psi) = m^*(\varphi) \cap m^+(\psi)$ and $m^*(\varphi) \cap m^-(\varphi \wedge \psi) = m^*(\varphi) \cap m^-(\psi)$. We only prove the former, as the latter can be proved analogously.

- ($\subseteq$): If both $x \in m^*(\varphi) \cap m^-(\varphi \wedge \neg\psi)$ and $x \notin m^*(\varphi) \cap m^+(\psi)$ hold, then there exists $S_1 \in CL(T)$ such that $x \models_{S_1} \varphi$ (from $x \in m^*(\varphi)$); and there also exists $S_2 \in CL(T)$ such that $x \models_{S_2} \neg\psi$ (from $x \notin m^+(\psi)$). Since A_φ and A_ψ are disjoint, we can construct an $S \in CL(T)$ such that
$$f'_i(x) = \begin{cases} f_i^1(x) & \text{if } i \in A_\varphi \\ f_i^2(x) & \text{if } i \in A_\psi, \end{cases}$$
where f_i^1, f_i^2, and f'_i are attribute functions of S_1, S_2, and S respectively. Hence, we have $x \models_S \varphi \wedge \neg\psi$, which contradicts $x \in m^-(\varphi \wedge \neg\psi)$ by Proposition 3.1.

- ($\supseteq$): If $x \in m^*(\varphi) \cap m^+(\psi)$, then for all $S \in CL(T)$, $x \models_S \psi$ holds, which implies that $x \not\models_S \varphi \wedge \neg\psi$. We then have $x \in m^*(\varphi) \cap m^-(\varphi \wedge \neg\psi)$ by Proposition 3.1.

■

Example 3.2 We use a modified example for route selection from [88] to illustrate POUDT and POUDL. The example is concerned with a route selection problem for solid waste management. In Table 2, six routes are evaluated by means of 2 attributes w (weight capacity) and s (surface condition), and the evaluation results are described by the decision attribute d. The domain of values for w is $V_w = \{l$ (low), m (medium), h (high)$\}$ and the domain of values for s is $V_s = \{v$ (very good), g (good), b (bad)$\}$. The evaluation results are then divided into three levels $V_d = \{1, 2, 3\}$. We assume that each domain is endowed with a preference relation such that $h \succ_w m \succ_w l$, $v \succ_s g \succ_s b$, and $3 \succ_d 2 \succ_d 1$, where $u \succ_i v$ means $u \succeq_i v \wedge v \not\succeq_i u$ for $i \in A$ and $u, v \in V_i$. Due to the incompleteness of the information, some attribute values appearing in the table are non-singleton. Note that this POUDT has 64 completions.

Let us now consider a POUDL wff $\varphi_1 = (\geq_w, m) \vee ((\leq_w, l) \wedge (\geq_s, g))$. The CNF of φ_1 is $\varphi_1^c = (w, V_w) \wedge ((\geq_w, m) \vee (\geq_s, g))$. We can see that $x_2 \models^+ \varphi_1$, since both $x_2 \models^+ (w, V_w)$ and $x_2 \models^+ (\geq_s, g)$ hold. Indeed, in any completion of T, we have either $f'_w(x_2) = l$ and $f'_s(x_2) = g$ or $f'_w(x_2) = m$ and $f'_s(x_2) = g$; therefore, $x_2 \models \varphi_1$ always holds. Note that $x_2 \models^+ \varphi_1$ can not be verified if we do not transform φ_1 into its CNF. Analogously, let φ_2 denote the POUDL wff

Table 3.1: A POUDT for route selection

$U \setminus A$	w	s	d
x_1	$\{m, h\}$	$\{v\}$	3
x_2	$\{l, m\}$	$\{g\}$	2
x_3	$\{m\}$	$\{g, v\}$	2
x_4	$\{l, m\}$	$\{b, g\}$	2
x_5	$\{l\}$	$\{b\}$	1
x_6	$\{h\}$	$\{g, v\}$	3

$(\leq_w, l) \wedge ((\geq_w, m) \vee (\leq_s, b))$, then its DNF is $\varphi_2^d = (w, \emptyset) \vee ((\leq_w, l) \wedge (\leq_s, b))$. It is easy to verify $x_2 \models^- \varphi_2$ by the semantics. Again, this cannot be done if φ_2 is not transformed into its CNF.

Let $\varphi = (\geq_w, m), \psi_1 = (\geq_d, 2)$, and $\psi_2 = (\geq_d, 3)$ be three wffs of POUDL, then we consider the rules $\varphi \longrightarrow \psi_1$ and $\varphi \longrightarrow \psi_2$. We have

$$\underline{m}(\varphi) = \{x_1, x_3, x_6\}, \;\; \overline{m}(\varphi) = \{x_1, x_2, x_3, x_4, x_6\}, \;\; m^*(\varphi) = \{x_2, x_4\},$$

$$\underline{m}(\varphi \wedge \psi_1) = \{x_1, x_3, x_6\}, \;\; \overline{m}(\varphi \wedge \psi_1) = \{x_1, x_2, x_3, x_4, x_6\},$$
$$m^+(\psi_1) = \{x_1, x_2, x_3, x_4, x_6\}, \;\; m^-(\psi_1) = \{x_5\},$$
$$\underline{m}(\varphi \wedge \psi_2) = \{x_1, x_6\}, \;\; \overline{m}(\varphi \wedge \psi_2) = \{x_1, x_6\},$$
$$m^+(\psi_2) = \{x_1, x_6\}, \;\text{ and }\; m^-(\psi_2) = \{x_2, x_3, x_4, x_5\}.$$

Thus, we obtain the asi, rsi, and cfi of these two rules as follows:

	asi	rsi	cfi
$\varphi \longrightarrow \psi_1$	$[3, 5]$	$[\frac{1}{2}, \frac{5}{6}]$	$[1, 1]$
$\varphi \longrightarrow \psi_2$	$[2, 2]$	$[\frac{1}{3}, \frac{1}{3}]$	$[\frac{2}{5}, \frac{2}{3}]$

It can be verified that these values indeed satisfy equalities in Proposition 3.2. Note that the rule $\varphi \longrightarrow \psi_1$ is strongly valid, whereas the rule $\varphi \longrightarrow \psi_2$ is neither weakly valid, nor strongly valid. ■

3.2.1 The running scenario

Let us continue the running scenario of the KTC recruitment process and conduct the basic competency test for the applicants. It is assumed that the basic competencies of an employee are related to his/her long-term performance. We assume the following set of basic competencies are tested: communication, planning, teamwork, and self-management [92]. The test result of each applicant is given in a 10-point score. To utilize the past experience, we retrieve the basic competency test results of the employees working in the company more than eight years and their average performance evaluation during those years. Unfortunately, because we consider employees who are recruited at least eight years ago, their basic competency test results may be partially missing. Thus, we have uncertain (or even missing) values for their test results. We still assume the performance of the employees are ranked into four categories: outstanding (A), good (B), fair

(C), and poor (D). By joining the data retrieved from the databases of basic competency test results and employees' performance evaluation, we obtain a POUDT with four decision attributes and one condition attribute. The attributes and their domains are coded as follows:

1. condition attributes:
   - c: communication competency, $V_c = \{0, 1, 2, \ldots, 10\}$;
   - p: planning competency, $V_p = \{0, 1, 2, \ldots, 10\}$;
   - t: teamwork competency, $V_t = \{0, 1, 2, \ldots, 10\}$;
   - s: self-management competency, $V_s = \{0, 1, 2, \ldots, 10\}$;
2. decision attribute:
   - d: average performance evaluation during the working years, $V_d = \{A, B, C, D\}$

It is assumed that V_c, V_p, V_t, and V_s are all endowed with the weak preference ordering $10 \succeq_i 9 \succeq_i \cdots \succeq_i 0$ and V_d is endowed with the ordering $A \succeq_d B \succeq_d C \succeq_d D$. As in the case of Example 2.3, we assume an appropriate data mining algorithm (e.g., the DRSA approach) is applied to derive POUDL rules. Also, it is still assumed that pre-determined thresholds for confidences and supports are used to select the decision rules. However, since the confidence and support of a POUDL rule are intervals instead of numbers, we chose the midpoint of an interval to represent its confidence or support. In this way, we obtain a set of POUDL rules that constitute the third knowledge base KB_3 of the recruitment process. A typical rule in KB_3 is like this

$$(c, \{7, 8, 9\}) \wedge (p, \{5, 6, 7\}) \wedge (t, \{9, 10\}) \wedge (s, \{8, 9\}) \longrightarrow (d, \{A\}).$$

Now, given the basic competency test result of an applicant, we can apply the POUDL rules to predict his/her long-term performance in the following way. Because there may be more than one rules in KB_3 whose antecedents can match the applicant's test result, we will apply all applicable rules to the applicant. Let x be an applicant and $\varphi \longrightarrow \psi$ is a POUDL rule in KB_2, then we will write $x \models \psi$ if the test result of x matches with φ. Considering all rules applicable to x, we denote $lp(x) = \cap\{s \mid x \models ((d, s)\}$. Intuitively, $lp(x)$ denotes the set of possible values of x's long-term performance prediction. Thus, after the long-term performance prediction phase, each applicant is associated with a set of grades denoting the prediction on his/her long-term performance. The process of this phase is shown in Figure 3.2, where the green part shows that the database may contain incomplete information.

3.3 Preference-ordered Fuzzy Data Tables

The preference-ordered fuzzy data table (POFDT) is a further generalization of POUDT. An approach for dealing with fuzzy information in PODT has been proposed in [32]. In this section, we propose an alternative based on our logical formalism. For any domain V, let $\mathbb{NF}(V)$ denote the set of all normalized fuzzy subsets of V. Recall that a fuzzy subset of domain V is normalized if $\sup_{x \in V} \mu(x) = 1$, where μ is the membership function of the fuzzy subset. A POFDT is a tuple

$$T = (U, A, \{(V_i, \succeq_i) \mid i \in A\}, \{f_i \mid i \in A\}),$$

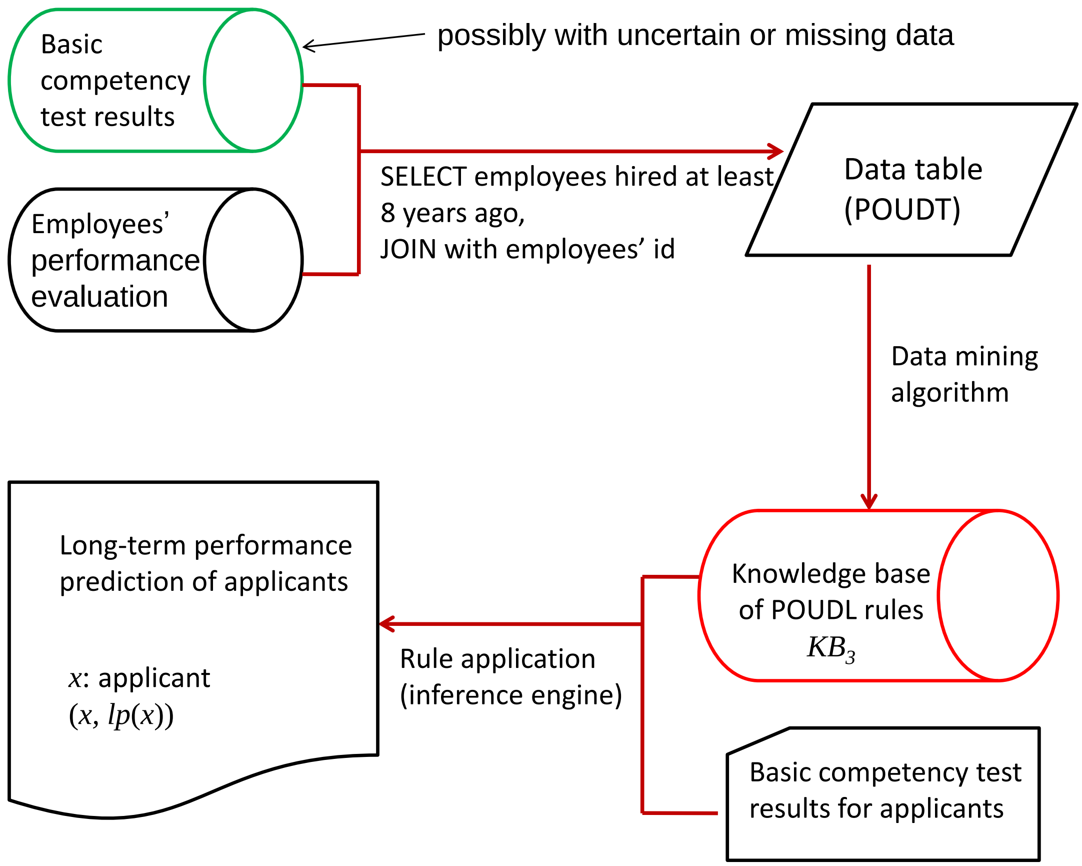


Figure 3.2: The long-term performance prediction phase of the recruitment process

where $U, A, \{(V_i, \succeq_i) \mid i \in A\}$ are defined as above, and for each $i \in A$, $f_i : U \rightarrow \mathbb{NF}(V_i)$.

For the representation of rules induced from POFDT, we can imagine several generalized decision languages, such as those introduced in [58, 59] and [16]. However, for simplicity, we use the syntax of PODL and interpret the wffs of PODL with respect to POFDT. Thus, the language of preference-ordered fuzzy decision logic (POFDL) is simply the language of PODL.

For the semantics of POFDL, we define the valuation function with respect to a POFDT over the wffs of POFDL. The function is denoted by E_T and defined by

1. $E_T(x, (\geq_i, v)) = \inf\{1 - \mu_i^x(v_i) \mid v_i \in V_i, v_i \not\succeq_i v\}$, where μ_i^x is the membership function of $f_i(x)$;

2. $E_T(x, (\leq_i, v)) = \inf\{1 - \mu_i^x(v_i) \mid v_i \in V_i, v_i \not\preceq_i v\}$, where μ_i^x is the membership function of $f_i(x)$;

3. $E_T(x, \neg\varphi) = 1 - E_T(x, \varphi)$;

4. $E_T(x, \varphi \wedge \psi) = E_T(x, \varphi) \otimes E_T(x, \psi)$, where $\otimes : [0, 1] \times [0, 1] \rightarrow [0, 1]$ is a t-norm[1]; and

[1] A binary operation $\otimes$ is a t-norm iff it is associative, commutative, and increasing in both places, and $1 \otimes a = a$

5. $E_T(x, \varphi \vee \psi) = E_T(x, \varphi) \oplus E_T(x, \psi)$, where $\oplus$ is the t-conorm defined by $a \oplus b = 1 - (1 - a) \otimes (1 - b)$.

Note that, if we consider the membership function μ_i^x as a possibility distribution on the domain V_i, $E_T(x, (\geq_i, v))$ corresponds to the necessity measure [96] of the subset $\{v_i \in V_i \mid v_i \succeq_i v\}$. The same remark holds for $E_T(x, (\leq_i, v))$.

The valuation function can be extended to cover the POFDL rules by the following equation

$$E_T(x, \varphi \longrightarrow \psi) = E_T(x, \varphi) \rightarrow_{\otimes} E_T(x, \psi),$$

where $\rightarrow_{\otimes}$: $[0,1] \times [0,1] \rightarrow [0,1]$ is the residuated implication function for $\otimes$, defined as $a \rightarrow_{\otimes} b = \sup\{x \mid x \otimes a \leq b\}$. As usual, we can omit the subscript T from E_T if this does not cause confusion.

The notions of validity, support, and confidence can be modified as follows.

Definition 3.2 *Let Φ_2 be the set of all POFDL rules and T be a POFDT. Then*

1. *the validity function $val_T : \Phi_2 \rightarrow [0,1]$ is*

$$val_T(\varphi \longrightarrow \psi) = \otimes_{x \in U} E_T(x, \varphi \longrightarrow \psi);$$

2. *the absolute support function $asp_T : \Phi_2 \rightarrow [0, |U|]$ is*

$$asp_T(\varphi \longrightarrow \psi) = \sum_{x \in U} E_T(x, \varphi \wedge \psi);$$

3. *the relative support function $rsp_T : \Phi_2 \rightarrow [0,1]$ is*

$$rsp_T(\varphi \longrightarrow \psi) = \frac{\sum_{x \in U} E_T(x, \varphi \wedge \psi)}{|U|};\ \text{and}$$

4. *the confidence function $cfd_T : \Phi_2 \rightarrow [0,1]$ is*

$$cfd_T(\varphi \longrightarrow \psi) = \frac{\sum_{x \in U} E_T(x, \varphi \wedge \psi)}{\sum_{x \in U} E_T(x, \varphi)}.$$

Example 3.3 We use a project evaluation system to illustrate POFDT and POFDL. Assume some projects are evaluated with respect to originality, presentation, and technical feasibility. The set of attributes is the same as in Example 2.1. The domains of values of these attributes are $[0, 9]$ endowed with ordinary ordering of real numbers. However, due to the difficulty of precise evaluation, the attribute values for these projects are fuzzy subsets represented by linguistic labels $\{a$ (excellent), b (good), c (fair), d (poor)$\}$. The membership functions of these subsets are given in Figure 3.3, and the evaluation results are presented in Table 3.

The membership functions are defined by

$$\mu_a(v) = \begin{cases} \frac{v-6}{3} & \text{if } 6 \leq v \leq 9 \\ 0 & \text{elsewhere} \end{cases}$$

---

and $0 \otimes a = 0$ for all $a \in [0,1]$.

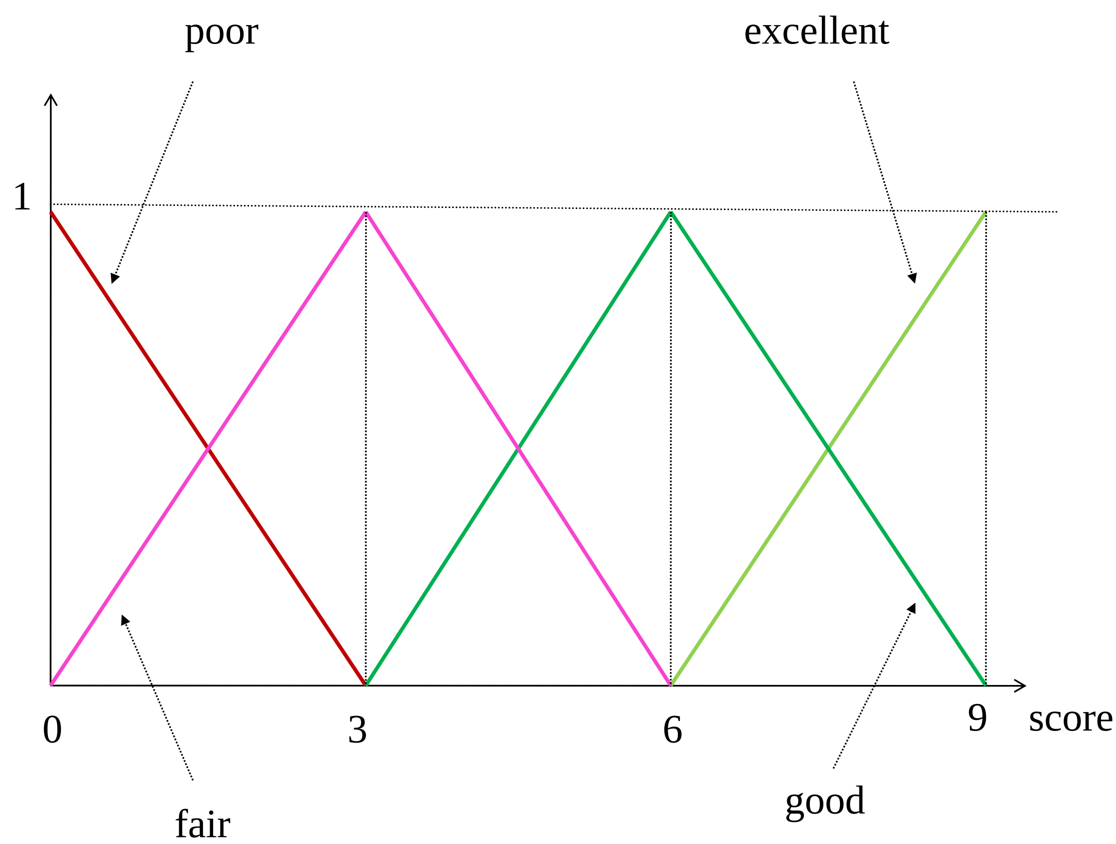


Figure 3.3: The membership functions of fuzzy sets

$$\mu_b(v) = \begin{cases} \frac{9-v}{3} & \text{if } 6 \le v \le 9 \\ \frac{v-3}{3} & \text{if } 3 \le v \le 6 \\ 0 & \text{elsewhere} \end{cases}$$

$$\mu_c(v) = \begin{cases} \frac{6-v}{3} & \text{if } 3 \le v \le 6 \\ \frac{v}{3} & \text{if } 0 \le v \le 3 \\ 0 & \text{elsewhere} \end{cases}$$

$$\mu_d(v) = \begin{cases} \frac{3-v}{3} & \text{if } 0 \le v \le 3 \\ 0 & \text{elsewhere} \end{cases}$$

Let us now consider three POFDL atomic formulas: $\varphi_1 = (\geq_o, 8)$, $\varphi_2 = (\geq_t, 4)$, and $\psi = (\geq_d, 6)$.

Table 3.2: A data table for project evaluation

$U \setminus A$	o	p	t	d
x_1	a	a	b	a
x_2	b	c	b	b
x_3	a	b	c	b
x_4	c	c	c	c
x_5	c	d	c	d
x_6	b	d	c	d
x_7	b	c	c	c
x_8	a	d	c	c

The truth value of an atomic formula for an object depends on the attribute values of the object as follows:

	$E_T(x, (\geq_i, 8))$	$E_T(x, (\geq_i, 4))$	$E_T(x, (\geq_i, 6))$
$f_i(x) = a$	1/3	1	1
$f_i(x) = b$	0	2/3	0
$f_i(x) = c$	0	0	0
$f_i(x) = d$	0	0	0

If we use the Gödel t-norm and implication [40] defined by

$$a \otimes b = \min(a, b), \text{ and}$$

$$a \rightarrow_\otimes b = \begin{cases} 1 & \text{if } a \le b \\ b & \text{if } a > b \end{cases},$$

we can obtain the truth values of the wffs of the objects as follows

	φ_1	φ_2	ψ	$\varphi_1 \wedge \varphi_2$	$\varphi_1 \wedge \varphi_2 \wedge \psi$	$\varphi_1 \wedge \varphi_2 \longrightarrow \psi$
x_1	1/3	2/3	1	1/3	1/3	1
x_2	0	2/3	0	0	0	1
x_3	1/3	0	0	0	0	1
x_4	0	0	0	0	0	1
x_5	0	0	0	0	0	1
x_6	0	0	0	0	0	1
x_7	0	0	0	0	0	1
x_8	1/3	0	0	0	0	1

Therefore, if r is the rule $\varphi_1 \wedge \varphi_2 \longrightarrow \psi$, we have $val_T(r) = 1, asp_T(r) = 1/3, rsp_T(r) = 1/24$, and $cfd_T(r) = 1$. ■

Note that, though POFDT is a generalization of POUDT, the quantitative measures of POFDL are not calculated through the completions of POFDT as in the case of POUDT. In fact, it is unclear how the notion of completion can be generalized to POFDT. We can envision at least two possibilities. One way is to consider a completion of a POFDT as a pair (T, c), where T is a classical DT and $c \in [0, 1]$; the other way is to define it as a pair (T, μ), where T is a classical DT and $\mu : Uni(T) \rightarrow [0, 1]$. Based on these definitions, we can derive the bounds of quantitative measures of POFDL rules as in Proposition 3.2. However, the detailed definitions and derivations of such results will be addressed in our future research.

3.3.1 The running scenario

Let us now proceed to the project proposal test phase of the KTC recruitment process. In this phase, the applicants are presented a case-study problem and requested to draft and submit a project proposal for solving the problem within a fixed time. The submitted proposal are evaluated with respect to its originality, presentation, and technical soundness and a score in $[0, 9]$ is given for each of these three aspects. On the other hand, the company has a database of real-world projects that have been carried out in KTC previously. Each project in the database has been evaluated with respect to these three aspects and the real degree of success. However, because it is in general difficult to give a precise score to a real-world project, only qualitative evaluation using linguistic terms like "excellent", "good", etc. is presented in the database. This means that the database is presented as a POFDT as that in Example 3.3. By using appropriate data mining algorithms, we can derive a set of POFDL rules. Also, it is still assumed that pre-determined thresholds for confidences and supports are used to select the decision rules, although the confidences and supports of these rules are calculated according to fuzzy logic semantics.

Consequently, we obtain a set of POFDL rules that constitute the forth knowledge base KB_4 of the recruitment process. Since the syntax of POFDL is exactly the same as that of PODL. The application of rules to the applicants' project proposal test results is the same as that in Example 3.1. Therefore, we will obtain an interval of possible scores $pp(x) = [pp_l(x), pp_u(x)]$ for each applicant after the project proposal test phase. The process of this phase is shown in Figure 3.4, where the green part shows that the database contains only qualitative (fuzzy) evaluations of past projects.

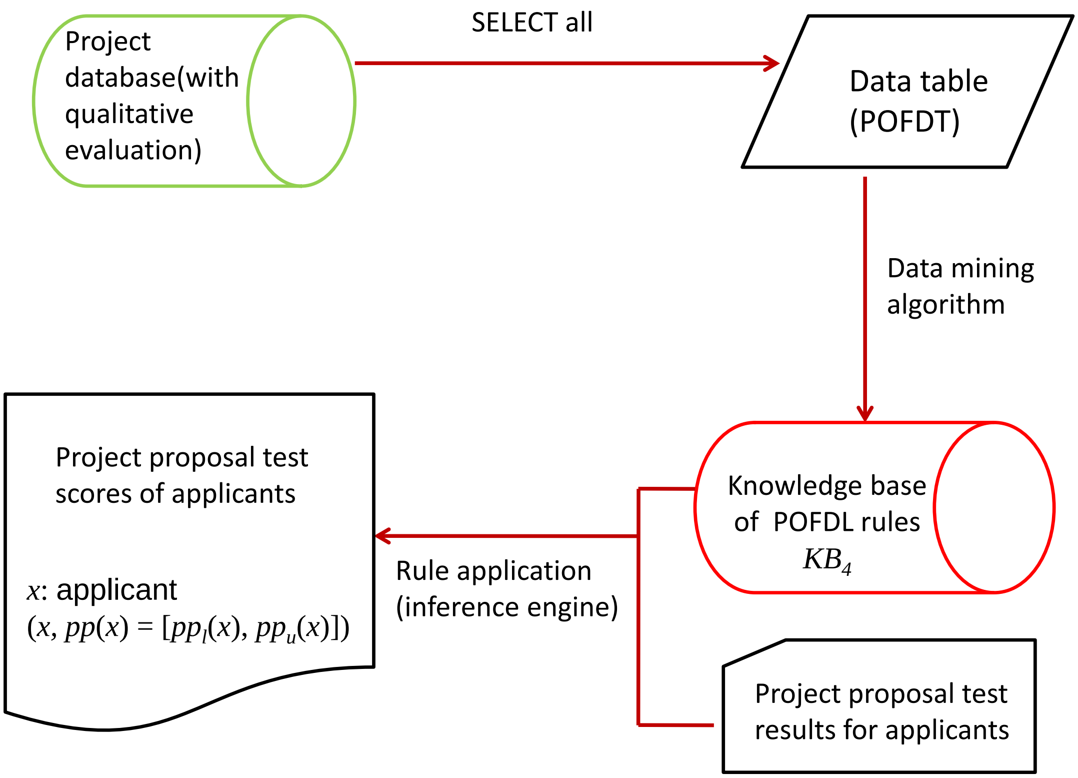


Figure 3.4: The project proposal test phase of the recruitment process

3.4 Pairwise Comparison Decision Logic

[29, 30, 31] proposed the pairwise comparison table (PCT) to handle multicriteria choice or ranking problems. In a PCT, the strength of preferences between objects, instead of the evaluation scores of objects, are stored with respect to each criterion. Formally, a PCT is a tuple

$$T = (U, A, \{H_i \mid i \in A\}, \{f_i \mid i \in A\}),$$

where U and A are defined as above; and for each $i \in A$, H_i is a finite set of integers, and $f_i : U \times U \to H_i$ encodes the preferential information[2]. Each H_i denotes a different grade of preference (such as "very weak preference", "weak preference", "strong preference", etc.) with respect to the criterion i. If $f_i(x, y) = h > 0$, then x is preferred to y by degree h with respect to the criterion i. If $f_i(x, y) = h < 0$, then x is not preferred to y by degree h with respect to the criterion i. If $f_i(x, y) = 0$, then x is similar to y with respect to the criterion i. A PCT is *coherent* if for each $i \in A$ and $x, y \in U$, $f_i(x, y) > 0$ implies $f_i(y, x) \leq 0$ and $f_i(x, y) < 0$ implies $f_i(y, x) \geq 0$. In this chapter, we only consider a coherent PCT.

[2] Without loss of generality, we slightly change the original definition in [29, 30, 31]

To represent rules induced from a PCT, we propose pairwise comparison decision logic (PCDL). An atomic formula of PCDL is a descriptor of the form $(i, \geq_h)$ or $(i, \leq_h)$, where $i \in A$ and $h \in H_i$, and the wffs and rules of PCDL are defined in the same way as those for the other decision logic languages discussed in this chapter. However, unlike other logics, where wffs are evaluated with respect to an object, the wffs of PCDL are evaluated with respect to a pair of objects. More precisely, the satisfaction of a wff with respect to a pair of objects (x, y) is defined as follows:

1. $(x, y) \models (i, \geq_h)$ iff $f_i(x, y) \geq h$,
2. $(x, y) \models (i, \leq_h)$ iff $f_i(x, y) \leq h$,
3. $(x, y) \models \neg\varphi$ iff $(x, y) \not\models \varphi$,
4. $(x, y) \models \varphi \wedge \psi$ iff $(x, y) \models \varphi$ and $(x, y) \models \psi$,
5. $(x, y) \models \varphi \vee \psi$ iff $(x, y) \models \varphi$ or $(x, y) \models \psi$.

If φ is a PCDL wff and T is a PCT, the set $m_T(\varphi)$ defined by:

$$m_T(\varphi) = \{(x, y) \in U \times U \mid (x, y) \models \varphi\} \tag{3.1}$$

is called the meaning set of the formula φ in T. If T is understood, we simply write $m(\varphi)$.

Definition 3.3 *Let Φ_3 be the set of all PCDL rules and $T = (U, A, \{H_i \mid i \in A\}, \{f_i \mid i \in A\})$ be a PCT. Then,*

1. *the rule $\varphi \longrightarrow \psi$ is valid in T iff $m_T(\varphi) \subseteq m_T(\psi)$;*
2. *the absolute support function $asp_T : \Phi_3 \to \aleph$ is*

$$asp_T(\varphi \longrightarrow \psi) = |m_T(\varphi \wedge \psi)|;$$

3. *the relative support function $rsp_T : \Phi_3 \to [0, 1]$ is*

$$rsp_T(\varphi \longrightarrow \psi) = \frac{|m_T(\varphi \wedge \psi)|}{|U|^2}; \text{ and}$$

4. *the confidence function $cfd_T : \Phi_3 \to [0, 1]$ is*

$$cfd_T(\varphi \longrightarrow \psi) = \frac{|m_T(\varphi \wedge \psi)|}{|m_T(\varphi)|}.$$

Note that the confidence function for PCDL rules was previously used to define the variable consistency model of a PCT [81].

Without loss of generality, we can rename the elements of U as natural numbers from 0 to $|U| - 1$. Then, each f_i can be seen as a $|U| \times |U|$ matrix M_i over domain H_i. Thus, we can employ matrix algebra to test the validity of a rule and calculate its support and confidence in an analogous way to that proposed in [55, 56].

By using PCDL, the three types of decision rules mentioned in [34] can be represented as follows:

1. $D_{\geq}$-decision rules:

$$\bigwedge_{i \in C} (i, \geq_{h_i}) \longrightarrow (d, \geq_1),$$

2. $D_{\leq}$-decision rules:

$$\bigwedge_{i \in C} (i, \leq_{h_i}) \longrightarrow (d, \leq_{-1}),$$

3. $D_{\geq\leq}$-decision rules:

$$\bigwedge_{i \in C_1} (i, \geq_{h_i}) \wedge \bigwedge_{i \in C_2} (i, \leq_{h_i}) \longrightarrow (d, \geq_1) \vee (d, \leq_{-1}),$$

where C, C_1, and $C_2 \subseteq A$ are sets of criteria, and $d \in A$ is the decision attribute. We assume that $\{-1, 1\} \subseteq H_d$ so that $f_d(x, y) = 1$ means that x outranks y, and $f_d(x, y) = -1$ means that y outranks x.

Example 3.4 Let us define a PCT from the PODT introduced in Example 3.1. The PCT is defined as

$$(U, A, \{H_i \mid i \in A\}, \{f'_i \mid i \in A\}),$$

where U and A are defined as in Example 2.1, $H_i = \{-3, -2, -1, 0, 1, 2, 3\}$, and f'_i is defined as $f'_i(x, y) = f_i(x) - f_i(y)$ for all $x, y \in U$ and $i \in A$, where f_i is also defined as in Example 2.1. Let us consider the following two rules:

$$r_1 = (o, \geq_2) \longrightarrow (d, \geq_1),$$

$$r_2 = (p, \leq_{-2}) \longrightarrow (d, \leq_0).$$

Then, we have

	asp	rsp	cfd
r_1	7	7%	$\frac{7}{8}$
r_2	15	15%	1

Note that the rule r_2 is valid, even though it only has a support value of 0.15. Furthermore, since in this example, $m((d, \geq_1) \vee (d, \leq_0)) = U \times U$ holds, the $D_{\geq\leq}$-decision rules are always valid and have a confidence value equal to 1. ■

3.4.1 The running scenario

We are now at the last phase of the KTC recruitment process—the decision phase. Up to now, we have three sets $sp(x), lp(x)$, and $pp(x)$ associated with each applicant, where $sp(x)$ and $lp(x)$ are both subsets of the set of ranks $\{A, B, C, D\}$, and $pp(x)$ is a sub-interval of the scores $[0, 9]$. The three sets indicate the potential of an applicant to some extent. We assume that such data for the present employees of the company is also kept and stored in the historical recruitment database. Furthermore, for the present employees, we have also the data of their actual performance evaluation. For any two employees x and y, we can compare their short-term or long-term performance prediction set by using the following definition:

$$spc(x, y) = |\{(v_1, v_2) \mid v_1 \in sp(x), v_2 \in sp(y), v_1 \succ_d v_2\}| - |\{(v_1, v_2) \mid v_1 \in sp(x), v_2 \in sp(y), v_2 \succ_d v_1\}|,$$

$lpc(x,y) = |\{(v_1, v_2) \mid v_1 \in lp(x), v_2 \in lp(y), v_1 \succ_d v_2\}| - |\{(v_1, v_2) \mid v_1 \in lp(x), v_2 \in lp(y), v_2 \succ_d v_1\}|$.

Intuitively, $spc(x,y)$ (or $lpc(x,y)$) is the number of possibilities that x is better than y minus the number of possibilities that y is better than x. The comparison of $pp(x)$ and $pp(y)$ is a bit complicated since they are continuous intervals instead of discrete sets. However, the basic principle is the same. To define $ppc(x,y)$, we simply consider a subset like $\{(v_1, v_2) \mid v_1 \in pp(x), v_2 \in pp(y), v_1 \succ_d v_2\}$ as an area on the two-dimensional space and compare these areas. An example is shown in Figure 3.5, where the possible score of x, denoted by $s(x)$, is in the interval $[l_x, u_x]$. In this example, $ppc(x,y)$ is equal to (the area of ABC−the area of $ACDE$). Furthermore,

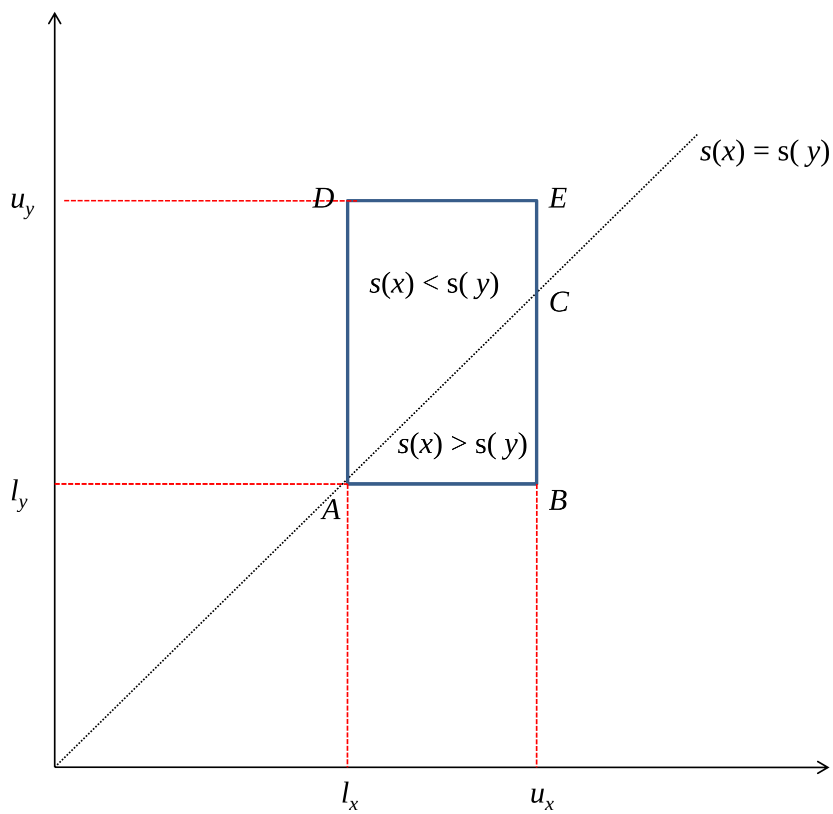


Figure 3.5: The comparison of possible scores $s(x)$ and $s(y)$

the performance evaluation of two employees can be compared and defined by $pec(x,y) = 1$ if x performs better than y, $pec(x,y) = -1$ if y performs better than x, and $pec(x,y) = 0$ if their performances are the same. Thus, from the historical recruitment database and employees' performance evaluation database, we can retrieved a PCT including three condition attributes spc, lpc, ppc and one decision attribute pec. By employing DRSA data mining approach to the PCT, we can derive a set of PCDL rules, which are used to constitute the fifth knowledge base KB_5 of the recruitment process. A typical rule in KB_5 may look like this:

$$(spc, \geq_4) \wedge (lpc, \geq_3) \wedge (ppc, \geq_2 0) \longrightarrow (pec, \geq_1).$$

Now, given the associated sp, lp, and pp sets for each applicant obtained in the previous phases, we can pairwisely compare any two applicants and find their corresponding spc, lpc, and ppc. The application of the rules in KB_5 will then result in a (possibly partial) ordering of the applicants. The ordering $\succ$ is derived for any two applicants x and y in the following way:

- if $(x, y) \models (pec, \geq_1)$, then $x \succ y$;
- if $(x, y) \models (pec, \geq_0)$, then $x \succeq y$;
- if $(x, y) \models (pec, \leq_0)$, then $y \succeq x$;
- if $(x, y) \models (pec, \leq_{-1})$, then $y \succ x$.

Based on such an ordering, the recruiters will interview with the applicants sequentially and make their final decision accordingly. The process of the final decision phase is shown in Figure 3.6.

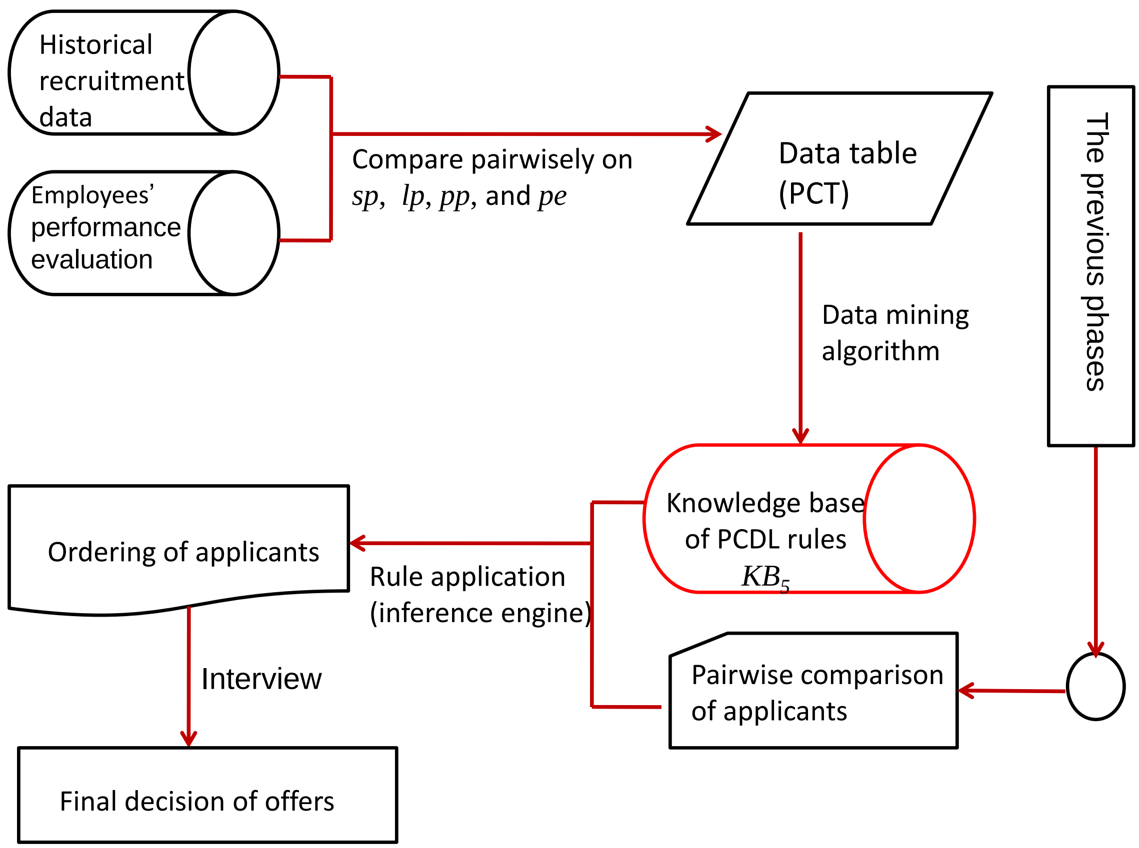


Figure 3.6: The final decision phase of the recruitment process

3.5 Summary and Remarks

In this chapter, we present some logics that are useful in the representation of rules induced from preference-ordered data tables. Such data tables are commonly used in MCDA. The main advantage of using logic is its syntax and semantics are precise. As DL is a precise way to represent decision rules induced from classical data tables, we use PODL and PCDL to reformulate the decision rules induced from PODT and PCDT in DRSA respectively. Though this seems a trivial step, it maps the decision rules induced from PODT and PCDT into precise logical formulas and gives them a formal semantics. The less trivial task is to generalize PODL to POUDL and POFDL. While the issue of missing values has been addressed in classical DT or PODT, we deal with uncertain values or fuzzy values. In particular, we derive the closed form for the lower and upper bounds of the confidence and support values of POUDL rules in each precise completion of the POUDT. We also present the semantics of POFDL rules based on possibility theory.

While this chapter is primarily concerned with the syntax and declarative semantics of the logics, efficient algorithms for data mining based on the logical representations are also urgently needed. Developing such algorithms is an important research direction.

In addition to decision logic, information logic is another kind of logic arising from data tables [14]. The semantics of information logic is the same as the Kripke semantics for modal logics. We believe that it would also be interesting to explore information logics with respect to dominance relations.

Chapter 4

Arrow Decision Logic for Relational Information Systems

In rough set theory, objects are partitioned into equivalence classes based on their attribute values, which are essentially functional information associated with the objects. Though many databases contain only functional information about objects, data about the relationships between objects has become increasingly important in decision analysis. A remarkable example is social network analysis, in which the principal types of data are attribute data and relational data.

To represent attribute data, a data table in rough set theory consists of a set of objects and a set of attributes, where each attribute is considered as a function from the set of objects to the domain of values for the attribute. Hence, such data tables are also called *functional information systems* (FIS), and rough set theory can be viewed as a theory of *functional granulation*. Recently, granulation based on relational information between objects, called *relational granulation*, has been studied by Liau and Lin [57]. To facilitate further study of relational granulation, it is necessary to represent and reason about data in *relational information systems* (RIS).

In FIS, the basic entities are objects, while DL formulas describe the properties of such objects, thus, the truth values of DL formulas are evaluated with respect to these objects. To reason about RIS, we need a language that can be interpreted in the domain of pairs of objects, since relations can be seen as properties of such pairs. Arrow logic (AL) [61, 87] fulfills this need perfectly. Hence, in this chapter, we propose arrow decision logic (ADL), which combines the main features of DL and AL, to represent the decision rules induced from RIS. The atomic formulas of ADL have the same descriptor form as those in DL; while the formulas of ADL are interpreted with respect to each pair of objects, just as in the pair frame of AL [61, 87]. The semantic models of ADL are RIS; thus, ADL can represent knowledge induced from systems containing relational information.

The remainder of this chapter is organized as follows. In Section 4.1, we give a precise definition of RIS. We study the relationship between these two kinds of information system and present some practical examples. In Section 4.2, we show how indiscernibility relations can be defined based on both FIS and RIS. In Section 4.3, we present the syntax and semantics of ADL. A complete axiomatization of ADL based on the combination of DL and AL axiomatic systems is presented. In Section 4.4, we define some quantitative measures for the rules of ADL and discuss the application of ADL to data analysis. Finally, we present our conclusions in Section 4.5.

4.1 Information Systems

Information systems are fundamental to rough set theory, in which the approximation space can be derived from attribute-value information systems [67]. In this section, we propose a generalization of functional information systems, namely, relational information systems. We present the algebraic relationship between these two kinds of information system, and several practical examples are employed to illustrate the algebraic notions.

4.1.1 Functional and relational information systems

We have seen that a data table is formally defined as an attribute-value information system and taken as the basis of the approximation space in rough set theory [67]. To emphasize the fact that each attribute in an attribute-value system is associated with a function on the set of objects, data tables are called *functional information systems* in this chapter. Thus, a functional information system (FIS) is simply a data table $T_f = (U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$.

In an FIS, the information about an object is consisted of the values of its attributes. Thus, given a subset of attributes $B \subseteq A$, we can define the information function associated with B as $Inf_B : U \to \prod_{i \in B} V_i$,

$$Inf_B(x) = (f_i(x))_{i \in B}. \tag{4.1}$$

Example 4.1 *One of the most popular applications in data mining is association rule mining from transaction databases [1, 41]. A transaction database consists of a set of transactions, each of which includes the number of items purchased by a customer. Each transaction is identified by a transaction id (tid). Thus, a transaction database is a natural example of an FIS, where*

- *U: the set of transactions, $\{tid_1, tid_2, \cdots, tid_n\}$;*
- *A: the set of possible items to be purchased;*
- *V_i: $\{0, 1, 2, \cdots, max_i\}$, where max_i is the maximum quantity of item i; and*
- *$f_i : U \to V_i$ describes the transaction details of item i such that $f_i(tid)$ is the quantity of item i purchased in tid.*

Though much information associated with individual objects is given in a functional form, it is sometimes more natural to represent information about objects in a relational form. For example, in a demographic database, it is more natural to represent the parent-child relationship as a relation between individuals, instead of an attribute of the parent or the children. In some cases, it may be necessary to use relational information simply because the exact values of some attributes may not be available. For example, we may not know the exact ages of two individuals, but we do know which one is older. These considerations motivate the following definition of an alternative kind of information system, called an RIS.

Definition 4.1 *A relational information system (RIS) is a quadruple*

$$T_r = (U, A, \{H_i \mid i \in A\}, \{r_i \mid i \in A\}),$$

where

- *U is a nonempty set, called the universe,*
- *A is a nonempty finite set of attributes,*
- *for each $i \in A$, H_i is the set of relational indicators for i, and*
- *for each $i \in A$, $r_i : U \times U \to H_i$ is a total function.*

A relational indicator in H_i is used to indicate the extent or degree to which two objects are related according to an attribute i. Thus, $r_i(x, y)$ denotes the extent to which x is related to y on the attribute i. When $H_i = \{0, 1\}$, then, for any $x, y \in U$, x is said to be i-related to y iff $r_i(x, y) = 1$.

Example 4.2 *Continuing with Example 2.1, assume that the reviewer is asked to compare the quality of the ten papers, instead of assigning scores to them. Then, we may obtain an RIS $T_r = (U, A, \{H_i \mid i \in A\}, \{r_i \mid i \in A\})$, where U and A are defined as in Example 2.1, $H_i = \{0, 1\}$, and $r_i : U \times U \to \{0, 1\}$ is defined by*

$$r_i(x, y) = 1 \Leftrightarrow f_i(x) \geq f_i(y)$$

for all $i \in A$.

4.1.2 Relationship between information systems

Before exploring the relationship between FIS and RIS, we introduce the notion of information system morphism (IS-morphism).

Definition 4.2

1. *Let $T_f = (U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$, and $T'_f = (U', A', \{V'_i \mid i \in A'\}, \{f'_i \mid i \in A'\})$ be two FIS; then an IS-morphism from T_f to T'_f is a $(|A| + 2)$-tuple of functions*

   $$\sigma = (\sigma_u, \sigma_a, (\sigma_i)_{i \in A})$$

   such that $\sigma_u : U \to U'$, $\sigma_a : A \to A'$ and $\sigma_i : V_i \to V_{\sigma_a(i)}$ $(i \in A)$ satisfy

   $$f'_{\sigma_a(i)}(\sigma_u(x)) = \sigma_i(f_i(x)) \tag{4.2}$$

   for all $x \in U$ and $i \in A$.

2. *Let $T_r = (U, A, \{H_i \mid i \in A\}, \{r_i \mid i \in A\})$, and $T'_r = (U', A', \{H'_i \mid i \in A'\}, \{r'_i \mid i \in A'\})$ be two RIS; then an IS-morphism from T_r to T'_r is a $(|A| + 2)$-tuple of functions*

   $$\sigma = (\sigma_u, \sigma_a, (\sigma_i)_{i \in A})$$

   such that $\sigma_u : U \to U'$, $\sigma_a : A \to A'$ and $\sigma_i : H_i \to H_{\sigma_a(i)}$ $(i \in A)$ satisfy

   $$r'_{\sigma_a(i)}(\sigma_u(x), \sigma_u(y)) = \sigma_i(r_i(x, y)) \tag{4.3}$$

   for all $x, y \in U$ and $i \in A$.

3. *If all functions in σ are 1-1 and onto, then σ is called an IS-isomorphism.*

An IS-morphism stipulates the structural similarity between two information systems of the same kind. Let T and T' be two such systems. Then we write $T \Rightarrow T'$ if there exists an IS-morphism from T to T', and $T \simeq T'$ if there exists an IS-isomorphism from T to T'. Note that $\simeq$ is an equivalence relation, whereas $\Rightarrow$ may be asymmetrical. Sometimes, we need to specify the properties of an IS-morphism. In such cases, we write $T \Rightarrow_{p_1,p_2} T'$ to indicate that there exists an IS-morphism σ from T to T' such that σ_u and σ_a satisfy properties p_1 and p_2 respectively. In particular, we need the notation $T \Rightarrow_{id,\text{onto}} T'$, which means that σ_u is the identity function of U (i.e., $\sigma_u(x) = id(x) = x$ for all $x \in U$) and σ_a is an onto function.

The relational information in an RIS may come from different sources. One of the most important sources may be the functional information. For various reasons, we may want to represent relational information between objects based on a comparison of some of the objects' attribute values. If all the relational information of an RIS is derived from an FIS, then it is said that the former is an embedment of the latter. Formally, this leads to the following definition.

Definition 4.3 *Let $T_f = (U, A_1, \{V_i \mid i \in A_1\}, \{f_i \mid i \in A_1\})$ be a FIS, and $T_r = (U, A_2, \{H_i \mid i \in A_2\}, \{r_i \mid i \in A_2\})$ be an RIS; then, an embedding from T_f to T_r is a $|A_2|$-tuple of pairs*

$$\varepsilon = ((B_i, R_i))_{i \in A_2},$$

where each $B_i \subseteq A_1$ is nonempty and each $R_i : \prod_{j \in B_i} V_j \times \prod_{j \in B_i} V_j \to H_i$ satisfies

$$r_i(x, y) = R_i(Inf_{B_i}(x), Inf_{B_i}(y)) \tag{4.4}$$

for all $x, y \in U$. T_r is said to be an embedment of T_f if there exists an embedding from T_f to T_r.

Note that the embedding relationship is only defined for two information systems with the same universe. Intuitively, T_r is an embedment of T_f if all relational informational in T_r is based on a comparison of some attribute values in T_f. Thus, for each attribute i in T_r, we can find a subset of attributes B_i in T_f such that the extent to which x is i-related to y is completely determined by comparing $Inf_{B_i}(x)$ and $Inf_{B_i}(y)$ in some particular way. We write $T_f \rhd T_r$ if T_r is an embedment of T_f.

Example 4.3 [**Pairwise comparison tables**] *Let T_f denote the FIS in Example 2.1, and $T_r = (U, A, \{H_i \mid i \in A\}, \{r_i \mid i \in A\})$, where $H_i = \{-3, -2, -1, 0, 1, 2, 3\}$, and r_i is defined as $r_i(x, y) = f_i(x) - f_i(y)$ for all $x, y \in U$ and $i \in A$. Then, the embedding from T_f to T_r becomes*

$$((\{o\}, R_o), (\{p\}, R_p), (\{t\}, R_t), (\{d\}, R_d)),$$

where $R_i : V_i \times V_i \to H_i$ is defined as

$$R_i(v_1, v_2) = v_1 - v_2$$

for all $i \in A$. The resultant T_r is an instance of the pairwise comparison table (PCT) used in MCDA [29, 30, 31, 33, 34, 35, 36, 82]. A similar embedment is used to define D-reducts (distance reducts) in [77], where a relationship between objects x and y exists iff the distance between $f_i(x)$ and $f_i(y)$ is greater than a given threshold.

Example 4.4 [**Dimension reduction and information compression**] *If $T_f = (U, A_1, \{V_i \mid i \in A_1\}, \{f_i \mid i \in A_1\})$ is a high dimensional FIS, i.e., $|A_1|$ is very large, then we may want to reduce the dimension of the information system. Furthermore, for security reasons, we may want to compress information in the FIS. An embedment based on rough set theory that can achieve both dimension reduction and information compression is as follows. First, the set of attributes, A_1, is partitioned into k mutually disjoint subsets, $A_1 = B_1 \cup B_2 \cup \cdots \cup B_k$, where k is substantially smaller than $|A_1|$. Second, for $1 \leq i \leq k$, define $R_i : \prod_{j\in B_i} V_j \times \prod_{j\in B_i} V_j \to \{0,1\}$ as $R_i(\mathbf{v}_i, \mathbf{v}'_i) = 1$ iff $\mathbf{v}_i = \mathbf{v}'_i$, where $\mathbf{v}_i, \mathbf{v}'_i \in \prod_{j\in B_i} V_j$. Thus, $((B_i, R_i)_{1\leq i\leq k})$ is an embedding from T_f to $T_r = (U, A_2, \{H_i \mid i \in A_2\}, \{r_i \mid i \in A_2\})$, where $A_2 = \{1, 2, \cdots, k\}$, $H_i = \{0, 1\}$, and $r_i(x, y) = 1$ iff $Inf_{B_i}(x) = Inf_{B_i}(y)$. Note that r_i is actually the characteristic function of the B_i-indiscernibility relation in rough set theory. Consequently, the dimension of the information system is reduced to k so that only the indiscernibility information with respect to some subsets of attributes is kept in the RIS.*

Example 4.5 [**Discernibility matrices**] *In [78], discernibility matrices are defined to analyze the complexity of many computational problems in rough set theory. This is especially useful in the computation of reduct in rough set theory. According to [78], given an FIS $T_f = (U, A_1, \{V_i \mid i \in A_1\}, \{f_i \mid i \in A_1\})$, its discernibility matrix is a $|U| \times |U|$ matrix D such that*

$$D_{xy} = \{i \in A_1 \mid f_i(x) \neq f_i(y)\}$$

for any $x, y \in U$. In other words, the (x, y) entry of the discernibility matrix is the set of attributes that can discern between x and y. More generally, we can define a discernibility matrix $D(B)$ with respect to any subset of attributes, $B \subseteq A_1$, such that

$$D(B)_{xy} = \{i \in B \mid f_i(x) \neq f_i(y)\}$$

for any $x, y \in U$. Let $B_1, \cdots, B_k$ be a sequence of subsets of attributes. Then, the sequence of discernibility matrices, $D(B_1) \cdots, D(B_k)$, can be combined as an RIS. The RIS becomes an embedment of T_f by the embedding $((B_i, R_i)_{1\leq i\leq k})$, such that $R_i : \prod_{j\in B_i} V_j \times \prod_{j\in B_i} V_j \to 2^{A_1}$ is defined by

$$R_i(\mathbf{v}_i, \mathbf{v}'_i) = \{j \in B_i \mid \mathbf{v}_i(j) \neq \mathbf{v}'_i(j)\},$$

where $\mathbf{v}(j)$ denotes the j-component of the vector $\mathbf{v}$.

Next, we show that the embedding relationship is preserved by IS-morphism transformation in some conditions. In the following theorem and corollary, we assume that T_f, T'_f, T_r, and T'_r have the same universe U. Thus,

$$T_f = (U, A_1, \{V_i \mid i \in A_1\}, \{f_i \mid i \in A_1\}),$$

$$T'_f = (U, A'_1, \{V'_i \mid i \in A'_1\}, \{f'_i \mid i \in A'_1\}),$$

$$T_r = (U, A_2, \{H_i \mid i \in A_2\}, \{r_i \mid i \in A_2\}),$$

$$T'_r = (U, A'_2, \{H'_i \mid i \in A'_2\}, \{r'_i \mid i \in A'_2\}).$$

Theorem 4.1

1. $T_f \rhd T_r$ *and* $T_r \Rightarrow_{id,\text{onto}} T'_r$ *implies* $T_f \rhd T'_r$.

2. $T_f \rhd T_r$ *and* $T'_f \Rightarrow_{id,\text{onto}} T_f$ *implies* $T'_f \rhd T_r$.

Proof: Let $\varepsilon = ((B_i, R_i))_{i\in A_2}$ be an embedding from T_f to T_r.

1. If $\sigma = (id, \sigma_a, (\sigma_i)_{i\in A_2})$ is an IS-morphism from T_r to T'_r such that σ_a is an onto function, then for each $j \in A'_2$, we can choose an arbitrary $i_j \in A_2$ such that $\sigma_a(i_j) = j$. Let B'_j and R'_j denote B_{i_j} and $\sigma_{i_j} \circ R_{i_j}$ respectively, then $\varepsilon' = ((B'_j, R'_j))_{j\in A'_2}$ is an embedding from T_f to T'_r. Indeed, by the definition of σ and ε, we have, for all $j \in A'_2$,

$$\begin{aligned} r'_j(x,y) &= r'_{\sigma_a(i_j)}(\sigma_u(x), \sigma_u(y)) && (\sigma_u = id, \sigma_a(i_j) = j)\\ &= \sigma_{i_j}(r_{i_j}(x,y)) && (Eq.\ 4.3)\\ &= \sigma_{i_j}(R_{i_j}(Inf_{B_{i_j}}(x), Inf_{B_{i_j}}(y))) && (Eq.\ 4.4)\\ &= R'_j(Inf_{B'_j}(x), Inf_{B'_j}(y))) && (R'_j = \sigma_{i_j} \circ R_{i_j}) \end{aligned}$$

2. If $\sigma = (id, \sigma_a, (\sigma_i)_{i\in A'_1})$ is an IS-morphism from T'_f to T_f such that σ_a is an onto function, then for each $j \in A_1$, we can choose an arbitrary $k_j \in A'_1$ such that $\sigma_a(k_j) = j$. For each $i \in A_2$, let $B'_i = \{k_j \mid j \in B_i\}$ and define $R'_i : \prod_{k_j\in B'_i} V'_{k_j} \times \prod_{k_j\in B'_i} V'_{k_j} \to H_i$ by

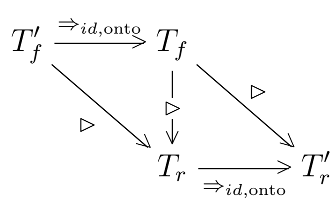
$$R'_i((v'_{k_j})_{j\in B_i}, (w'_{k_j})_{j\in B_i}) = R_i((\sigma_{k_j}(v'_{k_j}))_{j\in B_i}, (\sigma_{k_j}(w'_{k_j}))_{j\in B_i}). \tag{4.5}$$

Then, $\varepsilon' = ((B'_i, R'_i))_{i\in A_2}$ is an embedding from T'_f to T_r. This can be verified for all $i \in A_2$ as follows:

$$\begin{aligned} r_i(x,y) &= R_i(Inf_{B_i}(x), Inf_{B_i}(y)) && (Eq.\ 4.4)\\ &= R_i((f_j(x))_{j\in B_i}, (f_j(y))_{j\in B_i}) && (Eq.\ 4.1)\\ &= R_i((\sigma_{k_j}(f'_{k_j}(x)))_{j\in B_i}, (\sigma_{k_j}(f'_{k_j}(y)))_{j\in B_i}) && (Eq.\ 4.2)\\ &= R'_i((f'_{k_j}(x))_{k_j\in B'_i}, (f'_{k_j}(y))_{k_j\in B'_i}) && (Eq.\ 4.5; B'_i, \text{def.})\\ &= R'_i(Inf_{B'_i}(x), Inf_{B'_i}(y)) && (Eq.\ 4.1) \end{aligned}$$

■

The theorem can be represented by the following commutative diagram notation commonly used in category theory [5].



When the IS-morphism between two systems is an IS-isomorphism, we can derive the following corollary.

Corollary 4.1 *If* $T_f \rhd T_r$, $T_f \simeq T'_f$ *and* $T_r \simeq T'_r$, *then* $T'_f \rhd T'_r$

The commutative diagram of the corollary is:

$$\begin{array}{ccc} T_f & \xrightarrow{\rhd} & T_r \\ \simeq \big| & & \big| \simeq \\ T'_f & \xrightarrow[\rhd]{} & T'_r \end{array}$$

As shown in Example 4.4, an RIS may contain a summary of information about an FIS. Therefore, the RIS can serve as a tool for information summarization. If $T_f \rhd T_r$, then the information in T_r is less specific than that in T_f, i.e., the information is reduced. If as much information as possible is kept during the reduction, the embedding is called a trivial embedding. Formally, a *trivial embedding* from $T_f = (U, A_1, \{V_i \mid i \in A_1\}, \{f_i \mid i \in A_1\})$ to $T_r = (U, A_2, \{H_i \mid i \in A_2\}, \{r_i \mid i \in A_2\})$ is an embedding $\varepsilon = ((B_i, R_i))_{i \in A_2}$ such that each R_i is a 1-1 function. T_r is called a trivial embedment of T_f if there exists a trivial embedding from T_f to T_r. A trivial embedment plays a similar role to the initial algebra [11] in a class of RIS with the same attributes. This is shown in the next theorem, which easily follows from the definitions.

Theorem 4.2 *Let* $T_f = (U, A_1, \{V_i \mid i \in A_1\}, \{f_i \mid i \in A_1\})$, $T_r = (U, A_2, \{H_i \mid i \in A_2\}, \{r_i \mid i \in A_2\})$, *and* $T'_r = (U, A_2, \{H'_i \mid i \in A_2\}, \{r'_i \mid i \in A_2\})$ *be information systems. If* $\varepsilon = ((B_i, R_i))_{i \in A_2}$ *is a trivial embedding from* T_f *to* T_r *and* $\varepsilon' = ((B_i, R'_i))_{i \in A_2}$ *is an embedding from* T_f *to* T'_r*, then* $T_r \Rightarrow T'_r$.

Since embedding is an information reduction operation, many FIS may be embedded into the same RIS. Consequently, in general, it is not easy to recover an FIS that has been embedded into a given RIS. However, by applying the techniques of constraint solving, we can usually find possible candidates that have been embedded into a given RIS. More specifically, if the universe, the set of attributes, and the domain of values for each attribute of an FIS are known, then, given an embedding and the resultant embedded RIS, the problem of finding the FIS that is embedded into the given RIS is a constraint satisfaction problem (CSP). The following example illustrates this point.

Example 4.6 *Let* $U = \{1, 2, 3, 4, 5, 6\}$, $A = \{a, s\}$, $V_a = \{1, 2, \cdots, 120\}$, *and* $V_s = \{M, F\}$ *be, respectively, the universe, the set of attributes, and the domains of values for attributes* a *and* s*, where* a *denotes age and* s *denotes sex. Assume the RIS given in Table 4.1 results from an embedding* $((\{a\}, R_a), (\{s\}, R_s))$*, where* $R_i(v_1, v_2) = 1$ *iff* $v_1 = v_2$.

Then, to find an FIS $T_f = (U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$ *such that the RIS is the embedment of* T_f *by the above-mentioned embedding, we have to solve the following finite domain CSP, where* v_{ij} *is a variable denoting the value* $f_j(i)$ *to be found:*

$$\begin{array}{l} v_{ia} \in \{1, 2, \cdots, 120\}, v_{is} \in \{M, F\}, 1 \le i \le 6, \\ v_{1a} = v_{2a}, v_{3a} = v_{4a}, v_{5a} = v_{6a}, \\ v_{ia} \ne v_{ja}, (i, j) \ne (1, 2), (3, 4), \text{ or } (5, 6), \\ v_{1s} = v_{6s} \ne v_{2s} = v_{3s} = v_{4s} = v_{5s}. \end{array}$$

Table 4.1: An RIS obtained from a given embedding

a	1	2	3	4	5	6
1	1	1	0	0	0	0
2	1	1	0	0	0	0
3	0	0	1	1	0	0
4	0	0	1	1	0	0
5	0	0	0	0	1	1
6	0	0	0	0	1	1

s	1	2	3	4	5	6
1	1	0	0	0	0	1
2	0	1	1	1	1	0
3	0	1	1	1	1	0
4	0	1	1	1	1	0
5	0	1	1	1	1	0
6	1	0	0	0	0	1

4.2 General Theory of Granulation

In the preceding section, we showed that both FIS and RIS are useful formalisms for data representation. However, to achieve full generality, we can combine these two information systems. Let us define a hybrid information system (HIS) T as

$$(U, A \cup B, \{V_i \mid i \in A\}, \{H_i \mid i \in B\}, \{f_i \mid i \in A\}, \{r_i \mid i \in B\})$$

such that $(U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$ is an FIS and $(U, B, \{H_i \mid i \in B\}, \{r_i \mid i \in B\})$ is an RIS. In general, A and B are disjoint; however, this is not theoretically mandatory.

Since an HIS contains both functional and relational information, the indiscernibility relation defined in rough set theory must be generalized to accommodate both kinds of information. To present the general definition of an indiscernibility relation, we first generalize and rewrite the Kripke model for modal logic introduced in Chapter 1. In the current context, a Kripke model is a triple $(U, (R_i)_{i\in I}, (P_i)_{i\in J})$, where I and J are two sets of indices, $R_i \subseteq U \times U$ is a binary relation on U for $i \in I$, and $P_i \subseteq U$ is a subset of U for $i \in J$. The model introduced in Chapter 1 has only an accessibility relation. Here, we allow multiple accessibility relations and the relations are not necessarily equivalence relations, so the model is for polymodal logic. Furthermore, we use P_i to denote the set of possible worlds assigned to the primitive proposition p_i by the truth assignment π in the model of Chapter 1. Obviously, a Kripke model is an HIS in which the domains of the attributes and the relational indicators are all $\{0, 1\}$. On the other hand, we can transform an HIS into a Kripke model in the following way. Given $T = (U, A \cup B, \{V_i \mid i \in A\}, \{H_i \mid i \in B\}, \{f_i \mid i \in A\}, \{r_i \mid i \in B\})$, let $I = \{(i, h) \mid i \in B, h \in H_i\}$ and $J = \{(j, v) \mid j \in A, v \in V_j\}$, and define $R_{i,h}$ and $P_{j,v}$ as

$$(x, y) \in R_{i,h} \text{ iff } r_i(x, y) = h \text{ and } x \in P_{j,v} \text{ iff } f_j(x) = v$$

respectively. Then, $(U, (R_i)_{i\in I}, (P_i)_{i\in J})$ is a Kripke model corresponding to T. Thus, without loss of generality, we can develop a theory of granulation based on Kripke models, instead of general HIS.

To define the indiscernibility relation in a Kripke model $(U, (R_i)_{i\in I}, (P_i)_{i\in J})$, we need the notions of propositional and relational expressions. The sets of relational expressions (Π) and propositional expressions (Φ) based on I and J are defined by the following formation rules:

$$\Pi := i \mid \iota \mid \bar{\alpha} \mid \alpha^{\smile} \mid \alpha \sqcup \beta \mid \alpha \sqcap \beta \mid \alpha \cdot \beta,$$

where $i \in I$ and $\alpha, \beta \in \Pi$; and

$$\Phi := j \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \langle\alpha\rangle\varphi,$$

where $j \in J$, $\alpha \in \Pi$, and $\varphi, \psi \in \Phi$. For each $\alpha \in \Pi$, we can define the corresponding binary relation R_α on U recursively as follows:

$$\begin{aligned}
&R_\iota = \{(x,x) \mid x \in U\},\\
&R_{\bar{\alpha}} = \overline{R_\alpha} = U \times U - R_\alpha,\\
&R_{\alpha^\smile} = R_\alpha^\smile = \{(x,y) \mid (y,x) \in R_\alpha\},\\
&R_{\alpha\sqcup\beta} = R_\alpha \cup R_\beta,\\
&R_{\alpha\sqcap\beta} = R_\alpha \cap R_\beta,\\
&R_{\alpha\cdot\beta} = R_\alpha \cdot R_\beta = \{(x,y) \mid \exists z((x,z) \in R_\alpha \wedge (z,y) \in R_\beta)\};
\end{aligned}$$

and for each $\varphi \in \Phi$, we can define the corresponding proposition (subset) P_φ by

$$\begin{aligned}
&P_{\neg\varphi} = U - P_\varphi,\\
&P_{\varphi\vee\psi} = P_\varphi \cup P_\psi,\\
&P_{\varphi\wedge\psi} = P_\varphi \cap P_\psi,\\
&P_{\langle\alpha\rangle\varphi} = R_\alpha \cdot P_\varphi = \{x \mid \exists y((x,y) \in R_\alpha \wedge y \in P_\varphi)\}.
\end{aligned}$$

Given a Kripke model $(U, (R_i)_{i\in I}, (P_i)_{i\in J})$ and a propositional expression φ, the indiscernibility relation $ind(\varphi)$ can be defined easily, as in rough set theory, i.e.,

$$ind(\varphi) = \{(x,y) \in U \times U \mid x \in P_\varphi \Leftrightarrow y \in P_\varphi.\}.$$

However, defining an indiscernibility relation based on relational expressions is more complicated. Given a relational expression α, we define $ind(\alpha)$ iteratively. First, we define

$$E_0(\alpha) = \{(x,x) \mid x \in U\}.$$

Then, for $k > 0$, we define $E_k(\alpha)$ by the condition: $(x,y) \in E_k(\alpha)$ iff there exists a bijective (1-1 and onto) mapping $m : R_\alpha(x) \to R_\alpha(y)$ such that $(z, m(z)) \in E_{k-1}(\alpha)$ for all $z \in R_\alpha(x)$, where $R_\alpha(x) = \{z \mid (x,z) \in R_\alpha\}$ and $R_\alpha(y)$ is defined analogously. Finally, we define

$$ind(\alpha) = \bigcup_{k\geq 0} E_k(\alpha).$$

Next, we present some formal properties of an indiscernibility relation based on relational expressions. In the following discussion, we assume a fixed relational expression α. Thus, we write E_k and ind, instead of $E_k(\alpha)$ and $ind(\alpha)$.

Proposition 4.1 *For all $k \geq 0$, we have*

1. $E_k \subseteq E_{k+1}$.
2. *E_k is an equivalence relation in the sense that it satisfies*
   (a) *reflexivity: $(x,x) \in E_k$;*

(b) symmetry: if $(x, y) \in E_k$, then $(y, x) \in E_k$; and

(c) transitivity: if $(x, y), (y, z) \in E_k$, then $(x, z) \in E_k$

for all $x, y, z \in U$.

3. The indiscernibility relation ind is also an equivalence relation.

■

For each $x \in U$, let $[x]_k$ denote the E_k-equivalence class that contains x. Then, for each $X \subseteq U$, we define the multi-set[1] X/E_k as $\{[x]_k \mid x \in X\}$. According to the recursive definition of E_k, we have the following proposition:

Proposition 4.2 *For all $k > 0$ and $x, y \in U$, $(x, y) \in E_k$ iff $R_\alpha(x)/E_{k-1} = R_\alpha(y)/E_{k-1}$.* ■

The proposition provides an effective procedure for computing *ind* when U is finite. We start from E_0, and then calculate every E_k by using the previous equivalence relation E_{k-1} to find the multi-sets $R_\alpha(x)/E_{k-1}$ for each $x \in U$. The process is repeated until the condition $E_k = E_{k-1}$ is met. The final E_k is equal to *ind*.

Once the indiscernibility relation based on an expression (either propositional or relational) has been obtained, we can granulate the universe according to that relation. The granulation results in a partition of the universe into information granules, which are distinguished by relational or functional information about the objects.

4.3 Arrow Decision Logic

In the previous section, we demonstrated that FIS and RIS are useful formalisms for data representation. However, to represent and reason about knowledge extracted from information systems, we need a logical language. We have seen that decision logic (DL) can represent and reason about information in FIS (i.e. data tables).

Since relations can be seen as properties of pairs of objects, to reason about RIS, we need a language that can be interpreted in the domain of pairs of objects. Arrow logic (AL) language [61, 87] is designed to describe all things that may be represented in a picture by arrows. Therefore, it is an appropriate tool for reasoning about RIS.

To represent rules induced from an RIS, we propose arrow decision logic (ADL), derived by combining the main features of AL and DL. In this section, we introduce the syntax and semantics of ADL. The atomic formulas of ADL are the same as those in DL; while the formulas of ADL are interpreted with respect to each pair of objects, as in the pair frames of AL [61, 87].

4.3.1 Syntax and semantics of ADL

An atomic formula of ADL is a descriptor of the form (i, h), where $i \in A$, $h \in H_i$, A is a finite set of attribute symbols, and for each $i \in A$, H_i is a finite set of relational indicator symbols. In addition, the wffs of ADL are defined by the formation rules for AL, while the definition of derived connectives is the same as that in AL.

[1] A multi-set is a set in which a multiplicity of elements is allowed.

An interpretation of a given ADL is an RIS, $T_r = (U, A, \{H_i \mid i \in A\}, \{r_i \mid i \in A\})$, which can be seen as a square model of AL. Thus, the wffs of ADL are evaluated in terms of a pair of objects. More precisely, the satisfaction of a wff with respect to a pair of objects (x, y) in T_r is defined as follows (again, we omit the subscript T_r):

1. $(x, y) \models (i, h)$ iff $r_i(x, y) = h$,
2. $(x, y) \models \delta$ iff $x = y$,
3. $(x, y) \models \neg\varphi$ iff $(x, y) \not\models \varphi$,
4. $(x, y) \models \varphi \vee \psi$ iff $(x, y) \models \varphi$ or $(x, y) \models \psi$,
5. $(x, y) \models \otimes\varphi$ iff $(y, x) \models \varphi$,
6. $(x, y) \models \varphi \circ \psi$ iff there exists z such that $(x, z) \models \varphi$ and $(z, y) \models \psi$.

Let Σ be a set of ADL wffs; then, we write $(x, y) \models \Sigma$ if $(x, y) \models \varphi$ for all $\varphi \in \Sigma$. Also, for a set of wffs, Σ, and a wff, φ, we say that φ is an ADL consequence of Σ, written as $\Sigma \models \varphi$, if for every interpretation T_r and x, y in the universe of T_r, $(x, y) \models \Sigma$ implies $(x, y) \models \varphi$.

Example 4.7 *Let us use a RIS to represent the social network shown in Figure 4.1. The RIS is characterized by* $T_r = (U, A, \{H_i \mid i \in A\}, \{r_i \mid i \in A\})$, *where*

- $U = \{x_1, x_2, x_3, x_4\}$
- A =\{*Fr (Friend), Re (Recommendation), Tr (Trust)*\},
- $H_{\mathrm{Fr}} = H_{\mathrm{Tr}} = \{0, 1\}$, $H_{\mathrm{Re}} = \{0, 1, 2, 3\}$, *and*
- r_i *is shown in the three tables of the Figure.*

By using ADL semantics, we can see, for example,

$$(x_1, x_3) \models ((\mathrm{Fr}, 1) \circ (\mathrm{Re}, 3)) \wedge (\mathrm{Tr}, 1), \text{ and } (x_4, x_2) \models ((\mathrm{Fr}, 1) \circ (\mathrm{Re}, 2)) \wedge \neg(\mathrm{Tr}, 1).$$

Also, we have the following formulas valid in the RIS:

- $((\mathrm{Fr}, 1) \circ (\mathrm{Re}, 3)) \longrightarrow (\mathrm{Tr}, 1)$, *which means that if someone is strongly recommended by an agent's friend(s), then the agent will trust him,*
- $(\mathrm{Fr}, 1) \longrightarrow \otimes(\mathrm{Fr}, 1)$ *which means that friendship is symmetric, and*
- $(\mathrm{Fr}, 1) \longrightarrow (\mathrm{Tr}, 1) \wedge (\otimes(\mathrm{Tr}, 1))$, *which means that friends trust each other*

Fr	x_1	x_2	x_3	x_4
x_1	1	1	0	0
x_2	1	1	0	0
x_3	0	0	1	1
x_4	0	0	1	1

Re	x_1	x_2	x_3	x_4
x_1	0	0	0	1
x_2	0	0	3	0
x_3	0	2	0	0
x_4	1	0	0	0

Tr	x_1	x_2	x_3	x_4
x_1	1	1	1	0
x_2	1	1	1	0
x_3	0	1	1	1
x_4	0	0	1	1

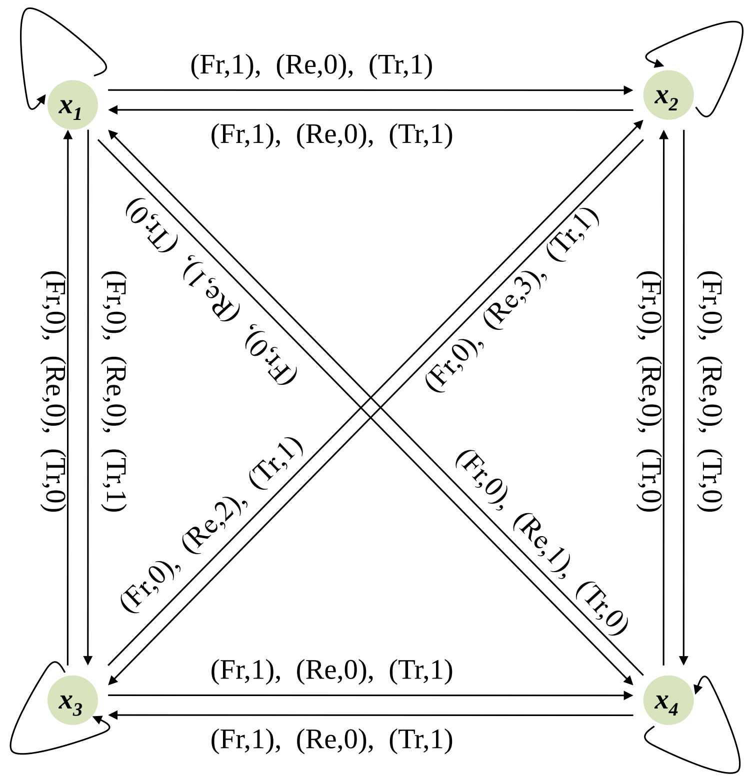


Figure 4.1: An ADL model

4.3.2 Axiomatization

The ADL consequence relation can be axiomatized by integrating the axiomatization of AL and specific axioms of DL. As shown in [61, 87], the AL consequence relations with respect to full square models can not be finitely axiomatized by an orthodox derivation system. To develop a complete axiomatization of AL, an unorthodox inference rule based on a difference operator $\mathtt{D}$ is added to the AL derivation system. The use of such unorthodox rules was first proposed by Gabbay [25]. The operator is defined in shorthand as follows:

$$\mathtt{D}\varphi = \top \circ \varphi \circ \neg\delta \vee \neg\delta \circ \varphi \circ \top,$$

where $\top$ denotes any tautology. According to the semantics, $\mathtt{D}\varphi$ is true in a pair (x, y) iff there exists a pair distinct from (x, y) such that φ is true in that pair.

The complete axiomatization of ADL consequence relations is presented in Figure 4.2, where $\varphi, \psi, \varphi', \psi'$, and χ are meta-variables denoting any wffs of ADL. The axiomatization consists of three parts: the propositional logic axioms; the DL and AL axioms; and the inference rules, including the classical Modus Ponens rule, the universal generalization rule for modal operators, and the unorthodox rule based on $\mathtt{D}$. The operator $\mathtt{D}$ is also utilized in DL3 to spread the axioms DL1 and DL2 to all pairs of objects. DL3 thus plays a key role in the proof of the completeness of the axiomatization. DL1 and DL2 are exactly the specific axioms of DL in [67]. An additional axiom

$$\neg(i,h) \equiv \bigvee_{h' \in H_i, h' \neq h} (i,h')$$

is presented in [67], but it is redundant. The AL axioms and inference rules can be found in [61], where AL4 is split into two parts and an extra axiom

$$\varphi \circ (\psi \circ (\delta \wedge \chi)) \supset (\varphi \circ \psi) \circ (\delta \wedge \chi)$$

is given. The extra axiom is called the *weak associativity axiom*, since it is weaker than the associativity axiom AL5. We do not need such an axiom, as it is an instance of AL5. The only novel axiom in our system is DL3. Though it is classified as a DL axiom, it is actually a connecting axiom between DL and AL.

An ADL derivation is a finite sequence $\varphi_1, \cdots, \varphi_n$ such that every φ_i is either an instance of an axiom or obtainable from $\varphi_1, \cdots, \varphi_{i-1}$ by an inference rule. The last formula φ_n in a derivation is called an ADL theorem. A wff φ is derivable in ADL from a set of wffs Σ if there are $\varphi_1, \ldots, \varphi_n$ in Σ such that $(\varphi_1 \wedge \ldots \wedge \varphi_n) \supset \varphi$ is an ADL-theorem. We use $\vdash \varphi$ to denote that φ is an ADL theorem and $\Sigma \vdash \varphi$ to denote that φ is derivable in ADL from Σ. Also, we write $\Sigma \vdash_{\mathrm{AL}} \varphi$ if φ is derivable in ADL from Σ without using the DL axioms.

The next theorem shows that the axiomatic system is sound and complete with respect to the ADL consequence relations.

Theorem 4.3 *For any set of ADL wffs* $\Sigma \cup \{\varphi\}$, *we have* $\Sigma \models \varphi$ *iff* $\Sigma \vdash \varphi$.

Proof:

1. Soundness: To show that $\Sigma \vdash \varphi$ implies $\Sigma \models \varphi$, it suffices to show that all ADL axioms are valid in any ADL interpretation and that the inference rules preserve validity. The fact that AL axioms are valid and the inference rules preserve validity follows from the soundness of AL, since any ADL interpretation is an instance of a square model. Furthermore, it is clear that DL axioms are valid in any ADL interpretation.

2. Completeness: To prove completeness, we first note that the set of instances of DL axioms is finite. Let us denote χ by the conjunction of all instances of DL axioms and χ_0 by the conjunction of all instances of axioms DL1 and DL2. If $\Sigma \not\vdash \varphi$, then $\Sigma \cup \{\chi\} \not\vdash_{\mathrm{AL}} \varphi$. Thus, by the completeness of AL, we have a pair model $\mathfrak{M} = (U \times U, C, R, I, \pi)$ and $x, y \in U$ such that $(x,y) \models_{\mathfrak{M}} \Sigma$, $(x,y) \models_{\mathfrak{M}} \chi$, and $(x,y) \models_{\mathfrak{M}} \neg\varphi$. Next, we show that $\mathfrak{M}$ can be transformed into an ADL interpretation. From $(x,y) \models_{\mathfrak{M}} \chi$, we can derive $(z,w) \models_{\mathfrak{M}} \chi_0$ for all $z, w \in U$ by DL3. Thus, for every $z, w \in U$ and $i \in A$, there exists exactly one $h_{i,z,w} \in H_i$ such that $(z,w) \models_{\mathfrak{M}} (i, h_{i,z,w})$. Consequently, $T_r = (U, A, \{H_i \mid i \in A\}, \{r_i \mid i \in A\})$, where $r_i(z,w) = h_{i,z,w}$ for $z, w \in U$ and $i \in A$, is an ADL interpretation such that $(x,y) \models_{T_r} \Sigma$ and $(x,y) \models_{T_r} \neg\varphi$. Thus,$\Sigma \not\vdash \varphi$ implies $\Sigma \not\models \varphi$. ■

4.4 Discussion and Applications

4.4.1 Discussion

Initially, it seems that an RIS is simply an instance of FIS whose universe consists of pairs of objects. However, there is a subtle difference between RIS and FIS. In FIS, the universe is an unstructured set, whereas in RIS, an implicit structure exists in the universe. The structure is made explicit by modal operators in ADL. For example, if (x, y) is a pair in an RIS, then (x, y) and (y, x) are considered to be two independent objects from the perspective of FIS; however, from the viewpoint of RIS, they are the converse of each other.

The difference between FIS and RIS is also reflected by the definition of IS-morphisms. When σ is an IS-morphism between two RIS, then, for a pair of objects (x, y), if (x, y) is mapped to (z, w), then (y, x) must be mapped to (w, z) at the same time. However, if these two RIS's are considered simply as FIS's with pairs of objects in their universes and σ is an IS-morphism between these two FIS's, then it is possible that $\sigma_u((x, y)) = (z_1, w_1)$ and $\sigma_u((y, x)) = (z_2, w_2)$ without $z_2 = w_1$ and/or $w_2 = z_1$. In other words, the images of (x, y) and (y, x) may be totally independent if we simply view an RIS as a kind of FIS. Therefore, even though FIS and RIS are very similar in appearance, they are mathematically and conceptually different.

In fact, FIS and RIS usually represent different aspects of the information about the objects. One of the main purpose of this chapter is to consider the relational structures of FIS. Sometimes, both functional and relational information about objects must be represented. Thus, to achieve full generality, we can combine these two kinds of information systems. Let us define a hybrid information system (HIS) as

$$(U, A \cup B, \{V_i \mid i \in A\}, \{H_i \mid i \in B\}, \{f_i \mid i \in A\}, \{r_i \mid i \in B\})$$

such that $(U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$ is an FIS and $(U, B, \{H_i \mid i \in B\}, \{r_i \mid i \in B\})$ is an RIS. Then, a HIS can represent functional and relational information about the same set of objects simultaneously. In general, A and B are disjoint; however, this is not theoretically mandatory.

The algebraic properties of IS-morphism between FIS was previously studied in [39] under the name of O-A-D homomorphism[2]. The notion of IS-morphism between RIS is a straightforward generalization of that between FIS. In fact, if FIS and RIS are considered as many-sorted algebras [6], both IS-morphism and O-A-D homomorphism can be seen as homomorphism in universal algebra [8, 11]. Indeed, we can consider information systems as a 3-sorted algebra whose sorts are the universe, the set of attributes, and the set of all attribute values (or relational indicators). Though homomorphism has been studied extensively in previous work, we define a novel notion of embedment between FIS and RIS to capture the relationship or transformation between two kinds of information related to the objects. This implies a new result, which shows that the embedding relationship can be preserved by IS-morphism under some conditions.

The investigation of RIS also facilitates a further generalization of rough set theory. In classical rough set theory, lower and upper approximations are defined in terms of indiscernibility relations based on functional information associated with the objects. However, it has been noted that many applications, such as social network analysis [74], need to represent both functional and relational information. Based on this observation, a concept of relational granulation was recently

[2]O, A, and D denotes objects, attributes, and the domain of values respectively.

proposed in [57]. The basic idea is that even though two objects are indiscernible with respect to their attribute values, they may still be discernible because of their relationships with other objects. Consequently, the definition of lower and upper approximations must consider the finer indiscernibility relations. In a future work, we will investigate different generalized rough sets based on the relational information associated with objects.

4.4.2 An Application of ADL to MCDM

Relational information plays an important role in MCDA. When rough set theory is applied to MCDA, it is crucial that preference-ordered attribute domains and decision classes be dealt with [34]. The original rough set theory could not handle inconsistencies arising from violation of the dominance principle due to its use of the indiscernibility relation. In previous work on MCDA, the indiscernibility relation was replaced by a dominance relation to solve the multi-criteria sorting problem, and the FIS was replaced by a PCT to solve multi-criteria choice and ranking problems [34]. A PCT is essentially an instance of an RIS, as shown in Example 4.3, in which the relations are confined to preference relations. This approach is called the *dominance-based rough set approach* (DRSA). For MCDA problems, DRSA can induce a set of decision rules from sample decisions provided by decision-makers. The induced decision rules play the role of a comprehensive preference model and can make recommendations about a new decision-making environment. The process whereby the induced decision rules are used to facilitate decision-making is called *multi-criteria decision making* (MCDM).

To apply ADL to MCDM, we define a rule of ADL as $\varphi \longrightarrow \psi$, where φ and ψ are wffs of ADL, called the antecedent and the consequent of the rule respectively. As in DL, let T_r be an interpretation of an ADL. Then, the set $m_{T_r}(\varphi)$ defined by

$$m_{T_r}(\varphi) = \{(x, y) \in U \times U \mid (x, y) \models \varphi\} \tag{4.6}$$

is called the *meaning set* of the formula φ in T_r. If T_r is understood, we simply write $m(\varphi)$. A wff φ is valid in T_r if $m(\varphi) = U \times U$. Some quantitative measures that are useful in data mining can be redefined for ADL rules.

Definition 4.4 *Let Φ be the set of all ADL rules and $T_r = (U, A, \{H_i \mid i \in A\}, \{r_i \mid i \in A\})$ be an interpretation of them. Then,*

1. *the rule $\varphi \longrightarrow \psi$ is valid in T_r iff $m_{T_r}(\varphi) \subseteq m_{T_r}(\psi)$*
2. *the absolute support function $\alpha_{T_r} : \Phi \to \mathbb{N}$ is*
$$\alpha_{T_r}(\varphi \longrightarrow \psi) = |m_{T_r}(\varphi \wedge \psi)|$$
3. *the relative support function $\rho_{T_r} : \Phi \to [0, 1]$ is*
$$\rho_{T_r}(\varphi \longrightarrow \psi) = \frac{|m_{T_r}(\varphi \wedge \psi)|}{|U|^2}$$
4. *the confidence function $\gamma_{T_r} : \Phi \to [0, 1]$ is*
$$\gamma_{T_r}(\varphi \longrightarrow \psi) = \frac{|m_{T_r}(\varphi \wedge \psi)|}{|m_{T_r}(\varphi)|}.$$

Without loss of generality, we assume that the elements of U are natural numbers from 0 to $|U|-1$. Each wff can then be seen as a $|U| \times |U|$ Boolean matrix, called its *characteristic matrix*. Thus, we can employ matrix algebra to test the validity of a rule and calculate its support and confidence in an analogous way to that proposed in [55, 56]. This is based on the intimate connection between AL and relation algebra [61, 87].

In the applications, we assume that the set of relational indicators H_i is a finite set of integers for every criterion i. Under such an assumption, we can use the following shorthand:

$$(i, \geq_{h_i}) \equiv \vee_{h \geq h_i}(i, h) \text{ and } (i, \leq_{h_i}) \equiv \vee_{h \leq h_i}(i, h).$$

By using ADL, the three main types of decision rules mentioned in [34] can be represented as follows:

1. $D_{\geq}$-decision rules:
$$\bigwedge_{i \in B}(i, \geq_{h_i}) \longrightarrow (d, \geq_1),$$
2. $D_{\leq}$-decision rules:
$$\bigwedge_{i \in B}(i, \leq_{h_i}) \longrightarrow (d, \leq_{-1}),$$
3. $D_{\geq\leq}$-decision rules:
$$\bigwedge_{i \in B_1}(i, \geq_{h_i}) \wedge \bigwedge_{i \in B_2}(i, \leq_{h_i}) \longrightarrow (d, \geq_1) \vee (d, \leq_{-1}),$$

where B, B_1, and $B_2 \subseteq A$ are sets of criteria and $d \in A$ is the decision attribute. We assume that $\{-1, 1\} \subseteq H_d$ so that $r_d(x, y) = 1$ means that x outranks y, and $r_d(x, y) = -1$ means that y outranks x.

Furthermore, the modal formulas of ADL allow us to represent some properties of preference relations. For example,

1. reflexivity: $\delta \longrightarrow (i, 0)$,
2. anti-symmetry: $\otimes(i, \geq_h) \longrightarrow (i, \leq_{-h})$, and
3. transitivity: $(i, \geq_{h_1}) \circ (i, \geq_{h_2}) \longrightarrow (i, \geq_{h_1+h_2})$.

Reflexivity means that each object is similar to itself in any attribute; anti-symmetry means that if x is preferred to y by degree (at least) h, then y is inferior to x by degree (at least) h; and transitivity denotes the additivity of preference degrees. The measures α, ρ, and γ can be used to assess the degree of reflexivity, anti-symmetry, and transitivity of an induced preference relation.

Example 4.8 *Let us continue to use the PCT in Example 4.3 and consider the following two ADL rules:*

$$s_1 = (o, \geq_2) \longrightarrow (d, \geq_1),$$
$$s_2 = (p, \leq_{-2}) \longrightarrow (d, \leq_0).$$

Then we have

	α	ρ	γ
s_1	7	0.07	0.875
s_2	15	0.15	1

Note that rule s_2 is valid even though it only has a support value of 0.15. Also, we observe that the anti-symmetry rule $\otimes(i, \geq_h) \longrightarrow (i, \leq_{-h})$ is valid in this PCT, which means that the preference relation is anti-symmetrical.

The main advantage of the ADL representation is its deduction capability. While many data mining algorithms for rule induction have been developed for MCDA, relatively little attention has been paid to the use of induced rules. We can consider two cases of using the rules to assist decision-making in real environments. In a simple situation, it suffices to match the antecedents of the rules with the real conditions. On the other hand, if the decision-maker encounters a complex situation, the deduction capability of ADL rules may be very helpful. For example, in a dynamic environment, the decision-maker has to derive a decision plan consisting of several decision steps, each of which may be guided by a different decision model. If each decision model is represented by a set of ADL wffs or rules, then the final decision can be deduced from the union of these sets and the set of ADL wffs representing the real conditions by our axiomatic deduction system. This is illustrated in Figure 4.3, where MCDA and MCDM are separated into two phases. In the MCDA phase, ordinary data mining algorithms are employed to find decision rules. This is the learning phase in which previous decision experiences are summarized into rules. In the MCDM phase, the rules are applied to the real environments. As indicated above, ADL representation plays an important role in this phase when the environment is highly complex.

4.4.3 An application of ADL to the representation of attribute dependency

One of the most important concepts of relational databases is that of functional dependency. In a relational schema design, functional dependencies are determined by the semantics of the relation, since, in general, they cannot be determined by inspection of an instance of the relation. That is, a functional dependency is a constraint, not a property derived from a relation. However, from a viewpoint of data mining, we can indeed find functional dependencies between attributes from a data table. Such dependencies are called *attribute dependencies*, the discovery of which is essential to the computation of reduct in rough set theory.

Let $T_f = (U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$ be an FIS and $B \cup \{i\} \subseteq A$ be a subset of attributes. Then, it is said that an attribute dependency between B and i, denoted by $B \longrightarrow i$, exists in T_f if for any two objects $x, y \in U$, $Inf_B(x) = Inf_B(y)$ implies $f_i(x) = f_i(y)$. Though the DL wffs can relate the values of condition attributes to those of decision attributes, the notion of attribute dependency can not be represented in DL directly. Instead, we can only simulate an attribute dependency between B and i by using a set of DL rules. More specifically, $B \longrightarrow i$ exists in T_f if for all $j \in B$ and $v_j \in V_j$, there exists $v \in V_i$ such that $\wedge_{j \in B}(j, v_j) \longrightarrow (i, v)$ is valid in T_f.

On the other hand, we can express an attribute dependency in an FIS by an ADL rule in an embedment of T_f. This embedding is a special instance of the embedding in Example 4.4. That is, we embed $T_f = (U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$ into the RIS $T_r = (U, A, \{H_i \mid i \in A\}, \{r_i \mid i \in$

$A\})$, where $H_i = \{0, 1\}$ and $r_i(x, y) = 1$ iff $f_i(x) = f_i(y)$. The embedding is $((i, R_i)_{i\in A})$, where, for every $i \in A$, $R_i : V_i \times V_i \to \{0, 1\}$ is defined by

$$R_i(v_i, v_i') = 1 \text{ iff } v_i = v_i'$$

for all $v_i, v_i' \in V_i$. Note that r_i is actually the characteristic function of the $\{i\}$-indiscernibility relation in rough set theory. Then, an attribute dependency $B \longrightarrow i$ in T_f can be represented as an ADL rule

$$\wedge_{j\in B}(j, 1) \longrightarrow (i, 1),$$

i.e., the attribute dependency $B \longrightarrow i$ exists in T_f iff the rule $\wedge_{j\in B}(j, 1) \longrightarrow (i, 1)$ is valid in T_r. If we view each pair of objects in $U \times U$ as a transaction and each attribute as an item in a transaction database, then an ADL rule of the form $\wedge_{j\in B}(j, 1) \longrightarrow (i, 1)$ is in fact an association rule. Thus, by transforming an FIS into its embedded RIS, the discovery of attribute dependencies can be achieved by ordinary association rule mining algorithms [1, 41].

4.4.4 The running scenario

In Chapters 2 and 3, we have shown that DL, PODL, POUDL, POFDL, and PCDL can serve as the underlying representation formalisms of the knowledge bases used in the recruitment process of the high-tech KMC. The human resource management department of KMC is responsible for not only the recruitment process but also suggestion of team formations for different projects in the company. It is usually desirable that the members in a team should be mutually trust to some extent. However, since the link of the new comers with the original employees is quite weak, the information about their relational network may be only partially known. In contrast, the information about the relational network of senior employees may be very complete. Social network analysis techniques can be applied to the relational network of the senior employees to uncover ADL rules like those in Example 4.7. Also, expertise of sociology may be imported into the knowledge base. In this way, we can constitute a knowledge base of *ADL* rules. Then, by using our axiomatic system, we can derive the trust relations between two employees based on partial information about their relational network. This kind of trust relationship is considered when a team is formed to ensure that teams members shall trust each other. The process is shown in Figure 4.4.

4.5 Summary and Remarks

In this chapter, we present ADL by combining DL and AL. ADL is useful for representing rules induced from an RIS. An important kind of RIS is PCT, which is commonly used in MCDA. The main advantage of using ADL is its precision in syntax and semantics. As DL is a precise way to represent decision rules induced from FIS, we apply ADL to reformulate the decision rules induced by PCT in DRSA. It is shown that such reformulation makes it possible to utilize ADL deduction in the highly complex decision-making process. We also show that ADL rules can represent attribute dependencies in FIS. Consequently, ordinary association rule mining algorithms can be used to discover attribute dependencies.

While this chapter is primarily concerned with the syntax and semantics of ADL, efficient algorithms for data mining based on logical representation are also important. In a future work, we will develop such algorithms.

In [79, 80], it is shown that the relations between objects are induced by relational structures in the domains of attributes. The preference relations in multicriteria decision-making and time windows in time series analysis are two important examples of relational structures. The derivation of the relations between objects from such structures is simply a kind of embedment. Certainly, we could consider structures beyond binary relational structures. Then, it should be possible to generalize RIS to non-binary relational information systems. We will also explore this possibility in our future work.

- Axioms:
  1. P: all tautologies of propositional calculus
  2. DL axioms:
     (a) DL1: $(i, h_1) \supset \neg(i, h_2)$, for any $i \in A$, $h_1, h_2 \in H_i$ and $h_1 \neq h_2$
     (b) DL2: $\vee_{h \in H_i}(i, h)$, for any $i \in A$
     (c) DL3: $\neg\mathtt{D}\neg\varphi_0$, where φ_0 is an instance of DL1 or DL2
  3. AL axioms:
     (a) AL0 (Distribution, DB):
        i. $(\varphi \supset \varphi')\underline{\circ}\psi \supset (\varphi\underline{\circ}\psi \supset \varphi'\underline{\circ}\psi)$
        ii. $\varphi\underline{\circ}(\psi \supset \psi') \supset (\varphi\underline{\circ}\psi \supset \varphi\underline{\circ}\psi')$
        iii. $\underline{\otimes}(\varphi \supset \psi) \supset (\underline{\otimes}\varphi \supset \underline{\otimes}\psi)$
     (b) AL1: $\varphi \equiv \otimes \otimes \varphi$
     (c) AL2: $\otimes(\varphi \circ \psi) \supset \otimes\psi \circ \otimes\varphi$
     (d) AL3: $\otimes\varphi \circ \neg(\varphi \circ \psi) \supset \neg\psi$
     (e) AL4: $\varphi \equiv \varphi \circ \delta$
     (f) AL5: $\varphi \circ (\psi \circ \chi) \supset (\varphi \circ \psi) \circ \chi$
- Rules of Inference:
  1. R1 (Modus Ponens, MP): $\dfrac{\varphi \quad \varphi \supset \psi}{\psi}$
  2. R2 (Universal Generalization, UG): $\dfrac{\varphi}{\underline{\otimes}\varphi} \qquad \dfrac{\varphi}{\varphi\underline{\circ}\psi} \qquad \dfrac{\varphi}{\psi\underline{\circ}\varphi}$
  3. R3 (Irreflexivity rule, $IR_{\mathtt{D}}$): $\dfrac{(p \wedge \neg\mathtt{D}p) \supset \varphi}{\varphi}$ provided that p is an atomic formula not occurring in φ

Figure 4.2: The axiomatic system for ADL

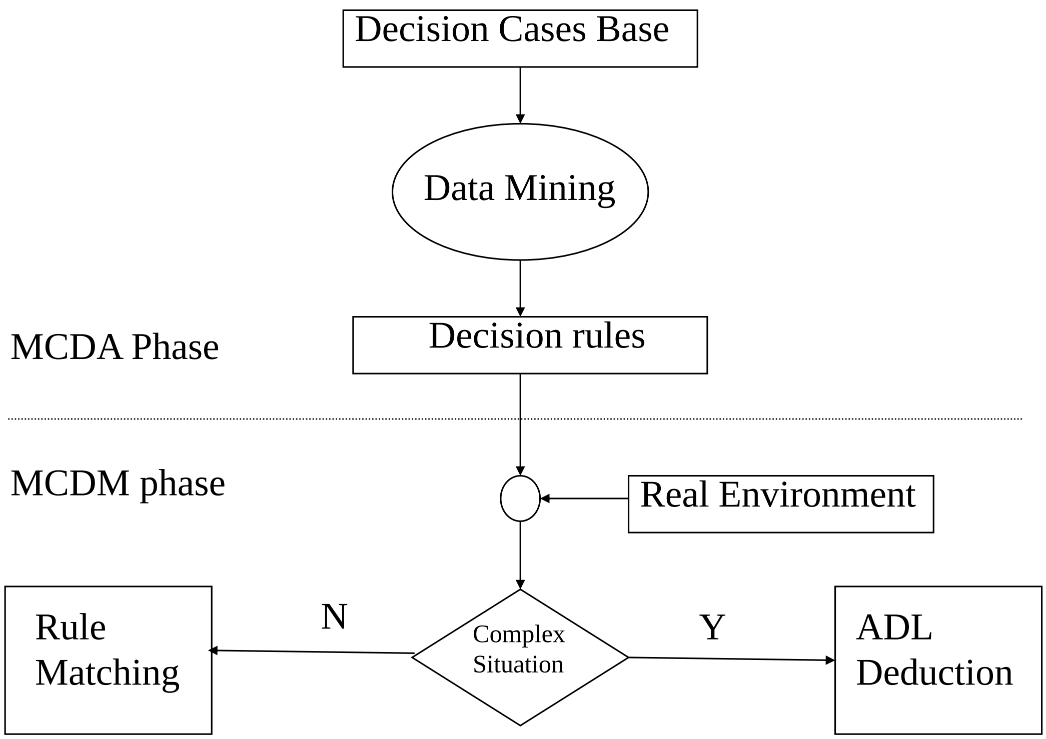


Figure 4.3: The MCDA and MCDM phases

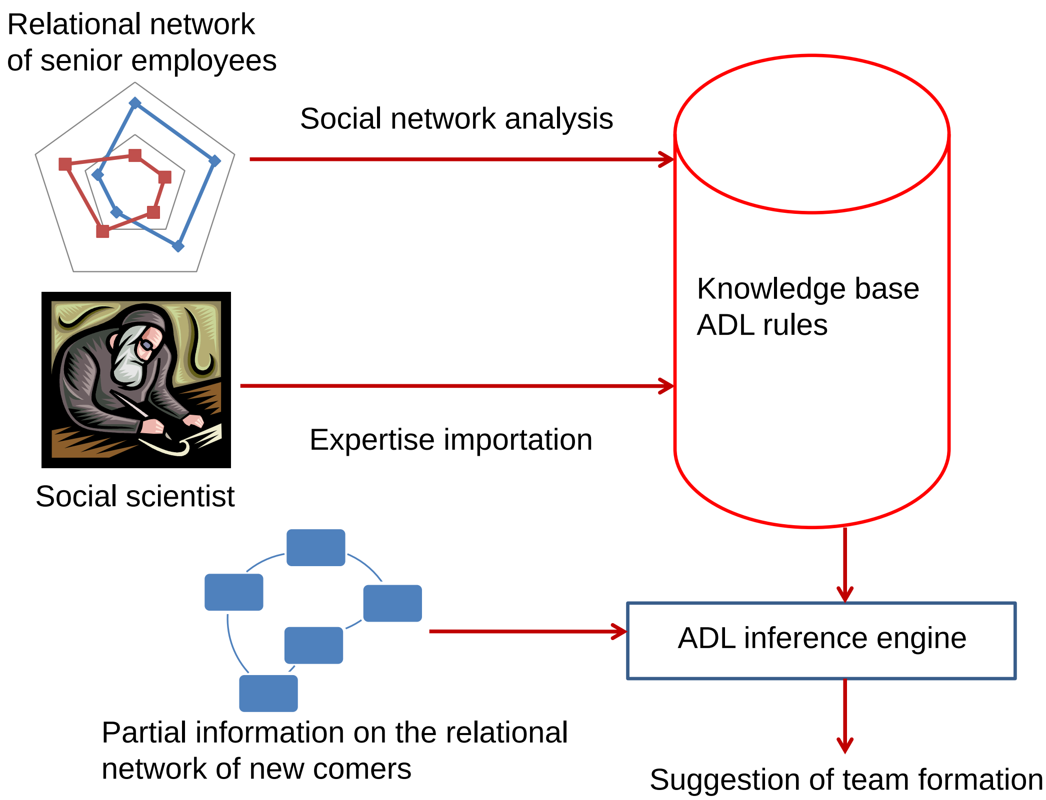


Figure 4.4: The process of team formation

Chapter 5

Conclusion

5.1 The Contribution of the Work

In this dissertation, we investigate several extensions of decision logic from the perspective of rough set theory. Traditionally, DL has been considered as a standard way of knowledge representation for rough set-based data mining, whereas our extensions show that DL-styled logics are also useful in more complicated knowledge management tasks.

On the one hand, we propose some decision logic languages for rule representation in rough set-based multicriteria decision analysis. The semantic models of these logics are data tables representing multicriteria decision records. Each decision record is described by a finite set of criteria/attributes. The domains of the criteria may have ordinal properties expressing preference scales, while the domains of the attributes may not.

On the other hand, we propose an arrow decision logic (ADL) to represent and reason about knowledge discovered from RIS. The logic combines the main features of decision logic (DL) and arrow logic (AL). AL is the basic modal logic of arrows. ADL formulas are interpreted in RIS which not only specifies the properties of objects, but also the relationships between objects. We present a complete axiomatization of ADL and discuss its application to knowledge representation in multicriteria decision analysis.

Our work is particularly useful for the knowledge representation phase in the knowledge management life cycle. A scenario about human resource management is used to show how the proposed logics can serve as representational formalisms in different stages of the recruitment process and team formation process of a company. The whole process of applications of our work in such a scenario is shown in Figure 5.1.

5.2 Remarks

In the last section of the thesis, we would like to discuss some issues related to DL and data mining that are not covered in preceding chapters.

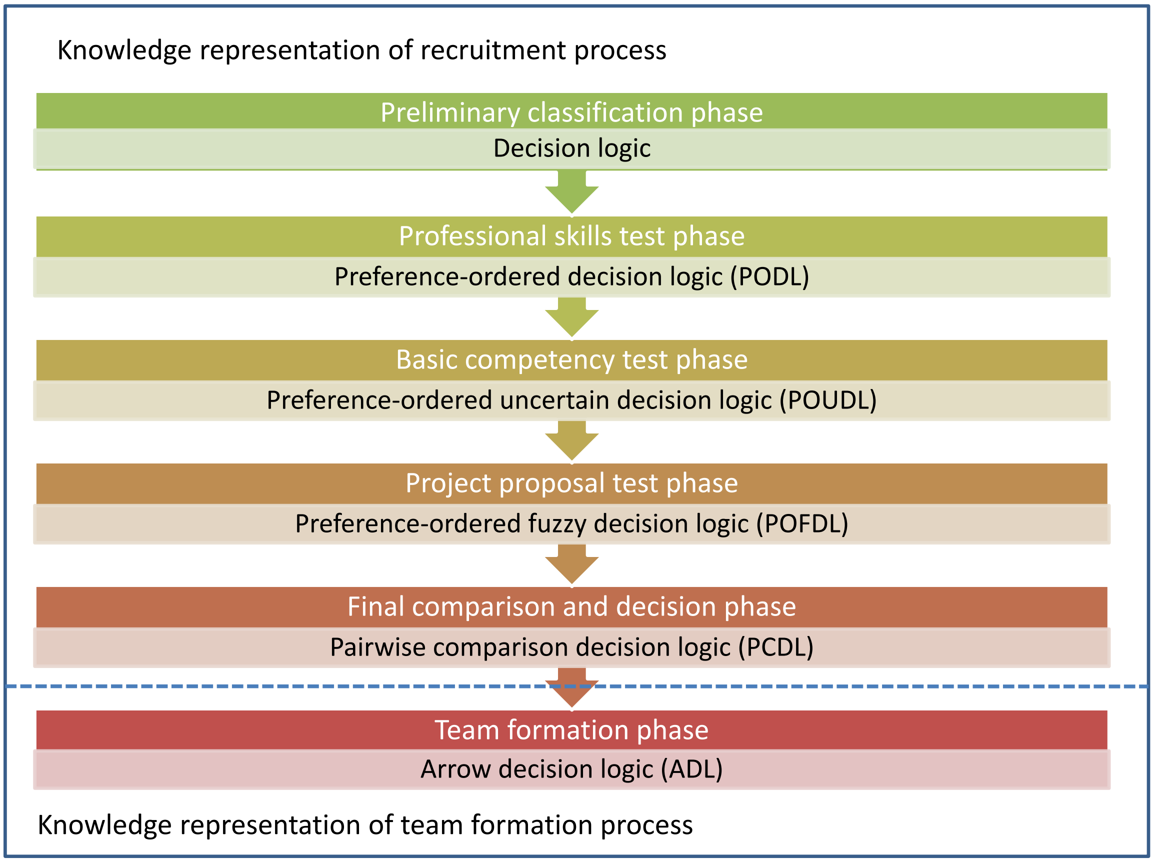


Figure 5.1: The process of the running scenario

5.2.1 A philosophical consideration

Induction is widely accepted as a cornerstone of the modern scientific methodology. It regards the systematic use of past experiences in the prediction of the future. According to Hume's epistemology, our beliefs are established on the basis of observation, and can be divided into observed matters of fact and unobserved matters of fact. While the beliefs of observed matters of fact are based directly on observation, unobserved matters of fact can only be known indirectly on the basis of observation by means of inductive argument.

An argument is called inductive if it passes from singular statements regarding the results of observations to universal statements, such as hypotheses or theories. The inductive argument can be formulated deductively as follows: Let $\varphi(x)$ denote a hypothesis regarding the individuals x we are interested in. In general, $\varphi(x)$ is in the form of a universal conditional sentence "$P(x) \supset Q(x)$", however it is not necessarily so. Let Φ and Ψ denote the following sentences

Φ : $\varphi(x)$ holds for all observed x.

Ψ : $\varphi(x)$ holds for all x (or at least the next observed x).

Then the inductive argument is just an application of the modus ponens rule and can be expressed as

$$\frac{\begin{array}{l}\Phi \\ \Phi \supset \Psi\end{array}}{\Psi} \tag{5.1}$$

where $\Phi \supset \Psi$ is called "The Principle of the Uniformity of Nature" (PUN).

If we can assume that the observation is noiseless, then the validity of the conclusion of the inductive argument will rely completely on that of PUN. However, since PUN represents unobserved portions of nature, it cannot be known directly by observation. Thus PUN, if known at all, must be known on the basis of induction. This means that to justify PUN, we must first justify the validity of the inductive argument which in turn relies on the validity of PUN. This kind of justification is thus circular and cannot serve as a proper justification. This leads to Hume's skepticism on induction [48].

In [17], it is shown that this kind of inductive skepticism also arises in the data mining context by considering the mining of rules from data tables in the framework of DL. Obviously, a data table consists of the observed matters of fact in some domain, whereas a mined rule is known indirectly on the basis of observation by means of inductive argument. Technically, it is shown that the justification of data mining schema leads to circularity or infinite regress in the framework of DL. Hence, a mined rule is simply a hypothesis from the philosophical viewpoint.

The above-mentioned result is by no means surprising since it has been argued that such circularity or infinite regress is inevitable in any attempt to the justification of induction. For example, one of the most important philosophers in the 20th century, Karl Popper, wrote in ([70], P.29):

> *...For the principle of induction must be a universal statement in its turn. Thus if we try to regard its truth as known from experience, then the very same problems which occasioned its introduction will arise all over again. To justify it, we should have to employ inductive inferences; and to justify these we should have to assume an inductive principle of a higher order; and so on. Thus the attempt to base the principle of induction on experience breaks down, since it must lead to an infinite regress.*

It seems that this argument also applies to data mining analogously according to our decision logic formulation.

This argument clearly shows the anti-inductivist position of Popper. It also leads him to refuse any form of verificationism of scientific theory, which claims that a scientific theory is supported by a collection of verifying instances. Instead, he proposes falsification as the methodology of science. In the falsification philosophy of science, a scientific theory is not confirmed by observational instances but corroborated by passing the test of possibly falsifying instances. As explained in [73]:

> *According to Popper, a scientific theory is posed without any prior justification. It gains its confirmation, or "corroboration", which is Popper's preferred term in later writings, not by experimental evidence demonstrating why it should hold, but by the fact that all attempts at showing why it does not hold have failed.*

In the 1960's, the verificationists have developed theories of confirmation [44, 9, 46]. These theories suggest a qualitative relation of confirmation between observations and scientific hypotheses or a quantitative measure for the degree of confirmation of a hypothesis by an observation. It seems that the most frequently used measures in data mining, i.e., the support and the confidence, are based on such verificationism philosophy. For example, it is not unusual in the data mining community to consider the confidence of a mined rule as the probability that the data confers upon the rule.

However, by the justification problem of induction, Popper claims that the attempt to develop a theory of confirmation for scientific theories is simply mistaken, so he proposed an alternative measure, called degree of corroboration, to compare competitive scientific hypotheses. Though we do not advocate such a radical position against verificationism philosophy, we indeed feel that data mining researchers can learn from the falsification methodology in the design of measures for mined rules which are conformed to the notion of corroboration. Therefore, our decision logic formulation for the justification problem of data mining will motivate the technical development of new kinds of measures for the strength of mined rules. A technical development of Popper's concept of corroboration for inductive inference has been carried out and applied to machine learning in [89]. It would be worthwhile to try the same thing in the data mining context. This kind of attempt represents a flow of ideas from the philosophy of science to data mining research.

In [90], the dynamic interactions between machine learning and the philosophy of science has been exemplified by the research of Bayesian networks, where a dynamic interaction between two disciplines is characterized by the flow of ideas in both directions. We have seen above that the justification problem of data mining may motivate the flow of ideas from the philosophy of science, in particular, the Popper's methodology, to data mining research, so we could also ask what impacts the data mining research has on the philosophy of science.

The answer to this question may not be too difficult if data mining is considered as a subdiscipline of machine learning. The viewpoint that machine learning is computational philosophy of science is not unusual. The ideas in the philosophy of science can be implemented and tested by machine learning algorithms, so machine learning techniques can help the philosophy of science to form new methodological rules. Let us quote the observation in [90]:

> *Machine learning has influenced the controversy in the philosophy of science between inductivism and falsificationism: Gillies argues that the success of the GOLEM machine learning program at learning a scientific hypothesis provides evidence for inductivism[27]. Proponents of falsificationism remain sceptical about machine learning[4], but further successes in the automation of scientific hypothesis generation may dampen their ardour. Machine learning has also proven invaluable to the philosophy of science as a means of testing formal models of scientific reasoning, including inductive reasoning[64, 23], abductive reasoning[15, 69, 24], coherence-based reasoning[84, 85, 86], analogical reasoning[47], causal reasoning, and theory revision[12].*

Of course, data mining can also help in the same way, though it is perhaps in a narrower scope. Therefore, a challenging research programme is to investigate to what extent data mining can serve as a computational tool for testing formal models of scientific reasoning in the philosophy of science.

Though it seems that the justification problem of data mining has a negative result, this does not render data mining useless. From a practical aspect, mined rules may not be justified scientific laws, but they may be very useful for a decision maker. Indeed, in many application domains, simply the summarization of the relationship between data is very helpful for knowledge management and decision making. In such cases, we are only concerned with the regularity within the scope of the existing data.

However, once we want to apply the mined rules beyond this scope, the justification problem could become critical. The result above implies that it is not justified to predict the future occurrence of an event by the regularity discovered from the past data. In fact, for a finite data table, a data mining algorithm can discover a large number of rules valid in that table. These rules may be mutually incompatible in the future instances to be observed. This kind of problem has been known as Goodman paradox or inductive consistencies in the philosophy of science[28, 45]. The implications of inductive consistencies for data mining will be discussed further in the extended version of this paper.

Last, but not the least, the skepticism on induction has lead Popper and the falsificationists to claim that all observations must be based on some background theory. In data mining context, this means that to discover interesting rules, we have to resort to a large amount of background knowledge. Background knowledge makes it possible to find not only the relations among the data but also the phenomena that give rise to data. This is exactly the direction of phenomenal data mining explored in [62].

5.2.2 Hypothesis selection

Since a mined rule is simply a hypothesis from the philosophical viewpoint, he natural followup question is then how to select and/or evaluate a hypothesis from a set of competitive ones. In [19], some possible criteria for hypothesis selection are presented based on the interpretation of data as models or as theories. In both interpretations, a logical language $\mathcal{L}$ (e.g. DL or its extensions presented in this thesis) is used for the representation of hypotheses (i.e. the mined rules).

Data as models

In this interpretation, each data item is considered as a model of the language $\mathcal{L}$. Hence, a hypothesis φ in $\mathcal{L}$ is verified or falsified in a data item d, denoted respectively by $d \models \varphi$ and $d \not\models \varphi$. In such interpretation, a data mining context is a pair $(\mathcal{L}, \mathcal{D})$, where $\mathcal{L}$ is the representation language for hypotheses and $\mathcal{D}$ is a finite set of models of $\mathcal{L}$.

A naive criterion for evaluating a hypothesis is the number (or proportion) of models verifying the hypothesis. In other words, we can select a hypothesis φ that maximizes $|m_{\mathcal{D}}(\varphi)|$ or $\frac{|m_{\mathcal{D}}(\varphi)|}{|\mathcal{D}|}$, where $m_{\mathcal{D}}(\varphi) = \{d \in \mathcal{D} \mid d \models \varphi\}$ is the meaning set of φ with respect to $\mathcal{D}$.

Someone may argue that not all data items in $\mathcal{D}$ is relevant to the evaluation of a hypothesis φ, so we should only concentrate on the relevant data items. This leads to the most popular criteria of hypothesis selection in data mining, i.e. support and confidence. Let us denote $d \not\perp \varphi$ if d is relevant to the evaluation of φ and define $r_{\mathcal{D}}(\varphi) = \{d \in \mathcal{D} \mid d \not\perp \varphi\}$, then the support and confidence of φ is respectively defined as

$$supp_{\mathcal{D}}(\varphi) = |r_{\mathcal{D}}(\varphi) \cap m_{\mathcal{D}}(\varphi)|$$

$$conf_{\mathcal{D}}(\varphi) = \frac{|r_{\mathcal{D}}(\varphi) \cap m_{\mathcal{D}}(\varphi)|}{|r_{\mathcal{D}}(\varphi)|}.$$

The usual practice in data mining is to set a threshold, and then evaluate the hypotheses with supports above the threshold according to their confidence values.

We must remark that the acceptance of a hypothesis based on a threshold suffers from the lottery paradox [54]. The resolution of the paradox and its implications on epistemology have been extensively discussed in philosophy [43, 42, 65], however, it seems that this problem has never received enough attention yet in data mining research.

Data as theories

In this interpretation, each data item is considered as a sentence of the language $\mathcal{L}$. Hence, a data set $\mathcal{D}$ is a finite subset of $\mathcal{L}$. We also assume that there is a consistent background theory $\mathcal{T} \subset \mathcal{L}$. Thus, a data mining context is a triplet $(\mathcal{L}, \mathcal{D}, \mathcal{T})$. We further assume a logical consequence relation $\models$ exists between sets of sentences and sentences of $\mathcal{L}$, so that $S \models \varphi$ means that $\varphi \in \mathcal{L}$ is a logical consequence of $S \subseteq \mathcal{L}$.

In such interpretation, a prerequisite for a hypothesis to be accepted is that it does not contradict with the background theory. That is, we consider a hypothesis φ only if $\mathcal{T} \not\models \neg\varphi$, where $\neg\varphi$ is the Boolean negation of φ. The sets of data items explained by φ, contradicted with φ, and neutral with φ are respectively defined as

$$e_{\mathcal{D}}(\varphi) = \{\alpha \in \mathcal{D} \mid \mathcal{T} \cup \{\varphi\} \models \alpha\},$$

$$c_{\mathcal{D}}(\varphi) = \{\alpha \in \mathcal{D} \mid \mathcal{T} \cup \{\varphi\} \models \neg\alpha\},$$

and

$$n_{\mathcal{D}}(\varphi) = \{\alpha \in \mathcal{D} \mid \mathcal{T} \cup \{\varphi\} \not\models \alpha \wedge \mathcal{T} \cup \{\varphi\} \not\models \neg\alpha\}.$$

Given these three sets, we can select hypotheses that explain the most data items and contradict the least data items. Sometimes, the importance of different data items may be different. If such information of importance is available, we can associate a weight to each data item in $\mathcal{D}$, and respectively define the explanatory and contradictory scores of a hypothesis φ as

$$es_{\mathcal{D}}(\varphi) = \frac{\sum_{\alpha \in e_{\mathcal{D}}(\varphi)} w_\alpha}{\sum_{\alpha \in \mathcal{D}} w_\alpha}$$

and

$$cs_{\mathcal{D}}(\varphi) = \frac{\sum_{\alpha \in c_{\mathcal{D}}(\varphi)} w_\alpha}{\sum_{\alpha \in \mathcal{D}} w_\alpha},$$

where w_α is the weight associated to a data item α.

To evaluate hypotheses, we can rank the hypotheses according to a lexicographical ordering on the pair $(es_{\mathcal{D}}(\varphi), 1 - cs_{\mathcal{D}}(\varphi))$, if the explanatory power of a hypothesis is emphasized, or a lexicographical ordering on the pair $(1 - cs_{\mathcal{D}}(\varphi), es_{\mathcal{D}}(\varphi))$, if the consistency property has the higher priority. Other intermediate combinations of these two criteria are possible by employing multi-criteria decision making techniques [49].

Alternative criteria

In addition to the criteria mentioned above, there are two general criteria for hypothesis selection, i.e. generality and simplicity.

1. Generality: a general hypothesis is usually preferred than the specific ones. In the logical sense, a hypothesis φ is more general than another hypothesis ψ if $\varphi \models \psi$ (or $\mathcal{T} \cup \{\varphi\} \models \psi$), but not vice versa.

2. Simplicity: for the economic purpose, a simple hypothesis is preferred than the complex ones. There is not a common agreement regarding the formal definition of simplicity yet. The criterion is related to the principle of the Occam's razor. A form of simplicity is formalized as *minimal description length principle* [71]. The space limit prevents us from presenting the technical detail of the principle. We refer the reader to a recent publication in this topic [38].

Furthermore, domain-specific criteria may be also helpful in the selection of hypotheses. For a data mining problem, these criteria of hypothesis selection may be in conflict with each other. Again, we can employ the techniques of multi-criteria decision making [49] to find overall optimal hypotheses.

5.3 Future Research

We have seen that DL can be used to describe knowledge discovered from a data table. In fact, DL is an instance of propositional logic where each descriptor (i, v) is a primitive proposition and each object in a data table is considered as an interpretation (a model) of the logic. Consequently, from the DL viewpoint, a data table is a set of models.

Alternatively, we can also describe data tables by using first-order logic (FOL) or many-sorted first-order logic (MSFOL). To describe data tables by FOL, we need only a fragment of it, called monadic function-free predicate logic (MFPL). Let us consider an instance of MFPL, called *first-order data logic* (FODL). The alphabet of FODL consists of the set of monadic predicate symbols $\{P_{i,v} \mid i \in A, v \in V_i\}$, (possibly) a set of constant symbols, Boolean connectives, and quantifiers. A data table $T = (U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$ is then considered as an interpretation of FODL $\mathfrak{A} = (U, \cdot^{\mathfrak{A}})$ such that the meaning of the predicate symbol $P_{i,v}$ is $\{u \in U \mid f_i(u) = v\}$.

On the other hand, a data table can be described by a MSFOL with a set of binary predicate symbols $\{P_i \mid i \in A\}$. The set of sorts of the logic is $\{\sigma_i \mid i \in A\} \cup \{\sigma_u\}$, so we call it *attribute value-sorted logic* (AVSL). The rank of a predicate symbol P_i is (σ_u, σ_i). A data table $T = (U, A, \{V_i \mid i \in A\}, \{f_i \mid i \in A\})$ is then considered as an interpretation of AVSL $\mathfrak{A} = (U, (V_i)_{i \in A}, \cdot^{\mathfrak{A}})$ such that the meaning of the predicate symbol P_i is $\{(u, v) \mid u \in U, v \in V_i, f_i(u) = v\}$. The syntax allows us to quantify not only on the objects but also on attribute values. This is particularly useful for the description of many-valued data tables, where $f_i(u)$ is a (not necessarily singleton) subset of V_i. In this case, the meaning of P_i is adapted to $\{(u, v) \mid u \in U, v \in V_i, v \in f_i(u)\}$. For example, we can then use $\forall x \exists y P_i(x, y)$ to express the fact that every object has non-null value on the attribute i. We believe that these alternative logical formulations of data tables deserve further exploration in the future.

Bibliography

[1] R. Agrawal and R. Srikant. Fast algorithm for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499. Morgan Kaufmann Publishers, 1994.

[2] R. Agrawal and R. Srikant. "Mining sequential patterns". In *Proc. 1995 Int. Conf. Data Engineering (ICDE95)*, pages 3–14, 1995.

[3] M. Alavi and D.E. Leidner. Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly*, 25(1):107–136, 2001.

[4] J.A. Allen. "Bioinformatics and discovery: induction beckons again". *BioEssays*, 23:104–107, 2001.

[5] A. Asperti and G. Longo. *Categories, Types, and Structures: An Introduction to Category Theory for the Working Computer Scientist.* MIT Press, 1991.

[6] G. Birkhoff and J. Lipson. Heterogeneous algebras. *Journal of Combinatorial Theory*, 8, 1970.

[7] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge Uinversity Press, 2001.

[8] S.N. Burris and H.P. Sankappanavar. *A Course in Universal Algebra.* Springer-Verlag, 1981.

[9] R. Carnap. *Logical Foundations of Probability.* The University of Chicago Press, Routledge, 1962.

[10] C.F. Chien and L.F. Chen. Using rough set theory to recruit and retain high-potential talents for semiconductor manufacturing. *IEEE Transactions On Semiconductor Manufacturing*, 20(4), 2007.

[11] P. M. Cohn. *Universal Algebra.* D. Reidel Publishing Co., 1981.

[12] L. Darden. "Anomaly-driven theory redesign: computational philosophy of science experiments". In T.W. Bynum and J.H. Moor, editors, *The Digital Phoenix: How Computers are Changing Philosophy*, pages 62–78. Blackwell Publishers, 1998.

[13] K. Dembczynski, S. Greco, and R. Slowinski. Methodology of rough-set-based classification and sorting with hierarchical structure of attributes and criteria. *Control & Cybernetics*, 31:891–920, 2002.

[14] S. Demri and E. Orlowska. *Incomplete Information: Structure, Inference, Complexity.* EATCS Monographs. Springer-Verlag, 2002.

[15] Y. Dimopoulos and A. Kakas. "Abduction and inductive learning". In L. De Raedt, editor, *Advances in Inductive Logic Programming*, pages 144–171. IOS Press, 1996.

[16] T.F. Fan, W.C. Hu, and C.J. Liau. Decision logics for knowledge representation in data mining. In *Proceedings of the 25th Annual International Computer Software and Applications Conference*, pages 626–631. IEEE Press, 2001.

[17] T.F. Fan and C.J. Liau. The justification problem of data mining—a decision logic formulation and its implications. In *Proc. of the IEEE ICDM2002 Workshop on the Foundation of Data Mining and Knowledge Discovery*, pages 113–118, 2002.

[18] T.F. Fan, C.J. Liau, D.R. Liu, and G.H. Tzeng. Granulation based on hybrid information systems. In *Proceedings of the 2006 IEEE International Conference on Systems, Man, and Cybernetics*, pages 4768–4772, 2006.

[19] T.F. Fan, D.R. Liu, and C.J. Liau. "Justification and hypothesis selection in data mining". In T.Y. Lin, S. Ohsuga, C.J. Liau, X. Hu, and S. Tsumoto, editors, *Foundations of Data Mining and Knowledge Discovery*, pages 119–130. Springer, 2005.

[20] T.F. Fan, D.R. Liu, and G.H. Tzeng. Arrow decision logic for relational information systems. *LNCS Transactions on Rough Sets*, V(LNCS 4100):240–262, 2006.

[21] T.F. Fan, D.R. Liu, and G.H. Tzeng. Rough set-based logics for multicriteria decision analysis. *European Journal of Operational Research*, 182(1):340–355, 2007.

[22] Z. Farkas. System integration via logic in knowledge management — a project initiative.

[23] P. Flach. *An Inquiry Concerning the Logic of Induction.* ITK Dissertation Series, 1995.

[24] P. Flach and A. Kakas. "On the relation between abduction and inductive learning". In D.M. Gabbay and R. Kruse, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems Vol. 4: Abductive Reasoning and Learning*, pages 1–33. Kluwer Academic Publishers, 2000.

[25] D.M. Gabbay. An irreflexivity lemma with applications to axiomatization of conditions on linear frames. In U. Mönnich, editor, *Aspects of Philosophical Logic*, pages 67–89. D. Reidel Publishing Co., 1981.

[26] F. Giannotti, G. Manco, D. Pedreschi, and F. Turini. "Experiences with a logic-based knowledge discovery support environment". In E. Lamma and P. Mello, editors, *AI*IA 99: Advances in Artificial Intelligence, 6th Congress of the Italian Association for Artificial Intelligence*, LNAI 1792, pages 202–213. Springer-Verlag, 1999.

[27] D. Gillies. *Artificial Intelligence and Scientific Method.* Oxford University Press, 1996.

[28] N. Goodman. *Fact, Fiction, and Forecast.* Bobbs-Merrill, Indianapolis, third edition, 1971.

[29] S. Greco, B. Matarazzo, and R. Slowinski. Rough set approach to multi-attribute choice and ranking problems. In *Proceedings of the 12th International Conference on Multiple Criteria Decision Making*, pages 318–329, 1997.

[30] S. Greco, B. Matarazzo, and R. Slowinski. Rough approximation of a preference relation in a pairwise comparison table. In L. Polkowski and A. Skowron, editors, *Rough Sets in Data Mining and Knowledge Discovery*, pages 13–36. Physica-Verlag, 1998.

[31] S. Greco, B. Matarazzo, and R. Slowinski. Rough approximation of a preference relation by dominance relations. *European Journal of Operational Research*, 117(1):63–83, 1999.

[32] S. Greco, B. Matarazzo, and R. Slowinski. The use of rough sets and fuzzy sets in MCDM. In T. Gal, T. Hanne, and T. Stewart, editors, *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory and Applications*, pages 14.1–14.59. Kluwer Academic Publishers, 1999.

[33] S. Greco, B. Matarazzo, and R. Slowinski. Extension of the rough set approach to multi-criteria decision support. *INFOR Journal: Information Systems and Operational Research*, 38(3):161–195, 2000.

[34] S. Greco, B. Matarazzo, and R. Slowinski. Rough set theory for multicriteria decision analysis. *European Journal of Operational Research*, 129(1):1–47, 2001.

[35] S. Greco, B. Matarazzo, and R. Slowinski. Rough sets methodology for sorting problems in presence of multiple attributes and criteria. *European Journal of Operational Research*, 138(2):247–259, 2002.

[36] S. Greco, B. Matarazzo, and R. Slowinski. Axiomatic characterization of a general utility function and its particular cases in terms of conjoint measurement and rough-set decision rules. *European Journal of Operational Research*, 158(2):271–292, 2004.

[37] S. Greco, B. Matarazzo, R. Slowinski, and J. Stefanowski. Variable consistency model of dominance-based rough set approach. In W. Ziarko and Y. Yao, editors, *Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing*, LNAI 2005, pages 170–181. Springer-Verlag, 2001.

[38] P. Grünwald, I.J. Myung, and M. Pitt. *Advances in Minimum Description Length: Theory and Applications*. The MIT Press, 2004.

[39] J.W. Grzymala-Busse. Algebraic properties of knowledge representation systems. In *Proceedings of the ACM SIGART International Symposium on Methodologies for Intelligent Systems*, pages 432–440. ACM Press, 1986.

[40] P. Hájek. *Metamathematics of Fuzzy Logic*. Kluwer Academic Publisher, 1998.

[41] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2001.

[42] J. Hawthorne. *Knowledge and Lotteries*. Oxford University Press, 2003.

[43] J. Hawthorne and L. Bovens. "The preface, the lottery, and rational belief". *Mind*, 108:241–264, 1999.

[44] C.G. Hempel. "Studies in the logic of confirmation". In C. Hempel, editor, *Aspects of Scienfific Explanation and Other Essays in the Philosophy of Science*, pages 3–46. Free Press, New York, 1965.

[45] C.G. Hempel. "Studies in the logic of confirmation". In C. Hempel, editor, *Aspects of Scienfific Explanation and Other Essays in the Philosophy of Science*, pages 53–79. Free Press, New York, 1965.

[46] J. Hintikka and P. Suppes. *Aspects of Inductive Logic*. North Holland, Amsterdam, 1967.

[47] D. Hofstadter and Fluid Analogies Research Group. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Penguin, 1998.

[48] D. Hume. *A Treatise of Human Nature*. Clarendon Press, Oxford, 1896.

[49] C.L. Hwang and K. Yoon. *Multiple Attribute Decision Making : Methods and Applications*. Springer-Verlag, 1981.

[50] R. Kruse, C. Borgelt, and D. Nauck. Fuzzy data analysis: challenges and perspectives. In *Proceedings of the 8th IEEE International Conference on Fuzzy Systems*, pages 1211–1216, San Francisco, CA, 1999. IEEE.

[51] M. Kryszkiewicz. Properties of incomplete information systems in the framework of rough sets. In L. Polkowski and A. Skowron, editors, *Rough Sets in Knowledge Discovery*, pages 422–450. Physica-Verlag, 1998.

[52] M. Kryszkiewicz and H. Rybiński. Reducing information systems with uncertain attributes. In Z. W. Raś and M. Michalewicz, editors, *Proceedings of the 9th ISMIS*, LNAI 1079, pages 285–294. Springer-Verlag, 1996.

[53] M. Kryszkiewicz and H. Rybiński. Reducing information systems with uncertain real value attributes. In *Proceedings of the 6th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*, pages 1165–1169, 1996.

[54] H. Kyburg. *Probability and the Logic of Rational Belief*. Wesleyan University Press, Middletown, 1961.

[55] C.J. Liau. Belief reasoning, revision and fusion by matrix algebra. In J. Komorowski and S. Tsumoto, editors, *Proceedings of the 4th International Conference on Rough Sets and Current Trends in Computing*, LNAI 3066, pages 133–142. Springer-Verlag, 2004.

[56] C.J. Liau. Matrix representation of belief states: An algebraic semantics for belief logics. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, 12(5):613–633, 2004.

[57] C.J. Liau and T.Y. Lin. Reasoning about relational granulation in modal logics. In *Proc. of the First IEEE International Conference on Granular Computing*, pages 534–538. IEEE Press, 2005.

[58] C.J. Liau and D.R. Liu. A logical approach to fuzzy data analysis. In J.M. Zytkow and J. Rauch, editors, *Proceedings of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery*, LNAI 1704, pages 412–417. Springer-Verlag, 1999.

[59] C.J. Liau and D.R. Liu. A possibilistic decision logic with applications. *Fundamenta Informaticae*, 46(3):199–217, 2001.

[60] W. Lipski. On databases with incomplete information. *Journal of the ACM*, 28(1):41–70, 1981.

[61] M. Marx and Y. Venema. *Multi-Dimensional Modal Logic*. Kluwer Academic Publishers, 1997.

[62] J. McCarthy. Phenomenal data mining. *Communications of the ACM*, 43(8):75–79, 2000.

[63] E. Mendelson. *Introduction to Mathematical Logic*. Chapman & Hall/CRC, forth edition, 1997.

[64] S. Muggleton and L. de Raedt. "Inductive logic programming: theory and methods". *Journal of Logic Programming*, 19/20:629–679, 1994.

[65] D.K. Nelkin. "The lottery paradox, knowledge, and rationality". *The Philosophical Review*, 109(3):373–409, 2000.

[66] I. Nonaka. A dynamic theory of organizational knowledge creation. *Organization Science*, 5(1):14–37, 1994.

[67] Z. Pawlak. *Rough Sets–Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, 1991.

[68] L. Polkowski, S. Tsumoto, and T.Y. Lin, editors. *Rough Set Methods and Applications*. Physica-Verlag, 2000.

[69] D. Poole. "Learning, Bayesian probability, graphical models, and abduction". In P. Flach and A. Kakas, editors, *Abduction and Induction: Essays on Their Relation and Integration*. Kluwer Academic Publishers, 1998.

[70] K. R. Popper. *The Logic of Scientific Discovery*. Harper & Row, New York, 1968.

[71] J. Rissanen. "Modeling by shortest data description". *Automatica*, 14:465–471, 1978.

[72] A. Satyadas, U. Harigopal, and N.P. Cassaigne. Knowledge management tutorial: an editorial overview. *IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews*, 31(4):429–437, 2001.

[73] P. Schroeder-Heister. “Popper, Karl Raimund”. In *International Encyclopedia of the Social and Behavioral Sciences*. Elsevier, 2001.

[74] J. Scott. *Social Network Analysis: A Handbook*. SAGE Publications, 2 edition, 2000.

[75] A. Skoworn and L. Polkowski, editors. *Rough Sets In Knowledge Discovery Vol 1: Methodology and Applications*. Physica-Verlag, 1998.

[76] A. Skoworn and L. Polkowski, editors. *Rough Sets In Knowledge Discovery Vol 2: Applications, Case Studies and Software Systems*. Physica-Verlag, 1998.

[77] A. Skowron. Extracting laws from decision tables. *Computational Intelligence: An International Journal*, 11:371–388, 1995.

[78] A. Skowron and C. Rauszer. The discernibility matrices and functions in information systems. In R. Slowinski, editor, *Intelligent Decision Support Systems: Handbook of Applications and Advances in Rough Set Theory*, pages 331–362. Kluwer Academic Publisher, 1991.

[79] A. Skowron and R. Swiniarski. Rough sets and higher order vagueness. In D. Slezak, G.Y. Wang, M. Szczuka, I. Duntsch, and Y.Y. Yao, editors, *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, LNAI 3641, pages 33–42. Springer-Verlag, 2005.

[80] A. Skowron and P. Synak. Complex patterns. *Fundamenta Informaticae*, 60(1-4):351–366, 2004.

[81] R. Slowinski, S. Greco, and B. Matarazzo. Mining decision-rule preference model from rough approximation of preference relation. In *Proceedings of the 26th IEEE Annual International Conference on Computer Software and Applications*, pages 1129–1134. IEEE Press, 2002.

[82] R. Slowinski, S. Greco, and B. Matarazzo. Rough set analysis of preference-ordered data. In J.J. Alpigini, J.F. Peters, A. Skowron, and N. Zhong, editors, *Proceedings of the 3rd International Conference on Rough Sets and Current Trends in Computing*, LNAI 2475, pages 44–59. Springer-Verlag, 2002.

[83] M.H. Stone. The theory of representations of boolean algebras. *Transactions of the American Mathematical Society*, 40:37–111, 1936.

[84] P. Thagard. *Computational Philosophy of Science*. The MIT Press, 1988.

[85] P. Thagard. *Conceptual Revolutions*. PrincetonUniversity Press, 1992.

[86] P. Thagard. “Probabilistic networks and explanatory coherence”. *Cognitive Science Quarterly*, 1:91–144, 2000.

[87] Y. Venema. A crash course in arrow logic. In M. Marx, L. P'olos, and M. Masuch, editors, *Arrow Logic and Multi-Modal Logic*, pages 3–34. CSLI Publications, 1996.

[88] R.H. Warren, J.A. Johnson, and G.H. Huang. Application of rough sets to environmental engineering. *LNCS Transactions on Rough Sets*, 1:356–374, 2004.

[89] P. Watson. "Inductive learning with corroboration". In O. Watanabe and T. Yokomori, editors, *Proceedings of the 10th International Conference on Algorithmic Learning Theory*, LNAI 1720, pages 145–156. Springer-Verlag, 1999.

[90] J. Williamson. Machine learning and the philosophy of science: a dynamic interaction. In *Proceedings of the ECML-PKDD-01 Workshop on Machine Leaning as Experimental Philosophy of Science*, 2001.

[91] T.D. Wilson. The nonsense of "knowledge management". *Information Research*, 8(1), 2002.

[92] W.W. Wu, Y.T. Lee, and G.H. Tzeng. Simplifying the manager competency model by using the rough set approach. In D.Slezak, J. Yao, J.F. Peters, W. Ziarko, and X. Hu, editors, *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, LNAI 3642, pages 484–494. Springer-Verlag, 2005.

[93] Y.Y. Yao and C.J. Liau. A generalized decision logic language for granular computing. In *Proceedings of the 11th IEEE International Conference on Fuzzy Systems*, pages 773–778. IEEE Press, 2002.

[94] Y.Y. Yao and Q. Liu. A generalized decision logic in interval-set-valued information tables. In N. Zhong, A. Skowron, and S. Ohsuga, editors, *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing*, LNAI 1711, pages 285–293. Springer-Verlag, 1999.

[95] Y.Y. Yao and N. Zhong. An analysis of quantitative measures associated with rules. In *Proceedings of the 2nd Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 479–488. IEEE Press, 1999.

[96] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3–28, 1978.