Chapter 2
Rough Set Theory

Abstract This chapter describes the foundations for rough set theory. We outline Pawlak's motivating idea and give a technical exposition. Basics of Pawlak's rough set theory and variable precision rough set model are presented with some related topics. We also present variants and related theories.

2.1 Pawlak's Approach

We begin with an exposition of Pawlawk's approach to rough set theory based on Pawlak [1]. His motivation is to provide a theory of knowledge and classification by introducing a new concept of set, i.e. *rough set*.

By *object*, we mean anything we can think of, for example, real things, states, abstract concepts, etc.. We can assume that knowledge is based on the ability to classify objects. Thus, knowledge is necessarily connected with the variety of classification patterns related to specific parts of the real or abstract world, called the *universe of discourse* (or the universe).

Now, we turn to a formal presentation. We assume the usual notation for set theory. Let U be non-empty finite set of objects called the *universe of discourse*. Any subset $X \subseteq U$ of the universe is called a *concept* or a *category* in U. Any family of concepts in U is called *knowledge* about U. Note that the empty set $\emptyset$ is also a concept.

We mainly deal with concepts which form a partition (classification) of a certain universe U, i.e. in families $C = \{X_1, X_2, ..., X_n\}$ such that $X_i \subseteq U$, $X_i \neq \emptyset$, $X_i \cap X_j = \emptyset$ for $i \neq j$, $i, j = 1, ..., n$ and $\bigcup X_i = U$. A family of classifications over U is called a *knowledge base* over U.

Classifications can be specified by using *equivalence relations*. If R is an equivalence relation over U, then U/R means the family of all equivalence classes of R (or classification of U) referred to as categories or concepts of R. $[x]_R$ denotes a category in R containing an element $x \in U$.

A *knowledge base* is defined as a relational system, $K = (U, \mathbf{R})$, where $U \neq \emptyset$ is a finite set called the universe, and $\mathbf{R}$ is a family of equivalence relations over U. $IND(K)$ means the family of all equivalence relations defined in K, i.e., $IND(K) = \{IND(\mathbf{P}) \mid \emptyset \neq \mathbf{P} \subseteq \mathbf{R}\}$. Thus, $IND(K)$ is the minimal set of equivalence relations,

S. Akama et al., *Reasoning with Rough Sets*, Intelligent Systems Reference Library 142, https://doi.org/10.1007/978-3-319-72691-5_2

containing all elementary relations of K, and closed under set-theoretical intersection of equivalence relations.

If $\mathbf{P} \subseteq \mathbf{R}$ and $\mathbf{P} \neq \emptyset$, then $\bigcap \mathbf{P}$ denotes the intersection of all equivalence relations belonging to $\mathbf{P}$, denoted $IND(\mathbf{P})$, called an *indiscernibility relation* of $\mathbf{P}$. It is also an equivalence relation, and satisfies:

$$[x]_{IND(\mathbf{P})} = \bigcap_{R \in \mathbf{P}} [x]_R.$$

Thus, the family of all equivalence classes of the equivalence relation $IND(\mathbf{P})$, i.e., $U/IND(\mathbf{P})$ denotes knowledge associated with the family of equivalence relations $\mathbf{P}$. For simplicity, we will write $U/\mathbf{P}$ instead of $U/IND(\mathbf{P})$.

$\mathbf{P}$ is also called $\mathbf{P}$-*basic knowledge*. Equivalence classes of $IND(\mathbf{P})$ are called *basic categories (concepts)* of knowledge $\mathbf{P}$. In particular, if $Q \in \mathbf{R}$, then Q is called a Q-*elementary knowledge* (about U in K) and equivalence classes of Q are referred to as Q-*elementary concepts (categories)* of knowledge $\mathbf{R}$.

Now, we describe the fundamentals of rough sets. Let $X \subseteq U$ and R be an equivalence relation. We say that X is R-*definable* if X is the union of some R-basic categories; otherwise X is R-undefinable.

The R-definable sets are those subsets of the universe which can be exactly defined in the knowledge base K, whereas the R-undefinable sets cannot be defined in K. The R-definable sets are called R-*exact sets*, and R-undefinable sets are called R-*inexact* or R-*rough*.

Set $X \subseteq U$ is called *exact* in K if there exists an equivalence relation $R \in IND(K)$ such that X is R-exact, and X is said to be *rough* in K if X is R-rough for any $R \in IND(K)$.

Observe that rough sets can be also defined *approximately* by using two exact sets, referred as a lower and an upper approximation of the set.

Suppose we are given knowledge base $K = (U, \mathbf{R})$. With each subset $X \subseteq U$ and an equivalence relation $R \in IND(K)$, we associate two subsets:

$$\underline{R}X = \bigcup \{Y \in U/R : Y \subseteq X\}$$
$$\overline{R}X = \bigcup \{Y \in U/R : Y \cap X \neq \emptyset\}$$

called the R-*lower approximation* and the R-*upper approximation* of X, respectively. They will be simply called the lower-approximation and the upper-approximation if the context is clear.

It is also possible to define the lower and upper approximation in the following two equivalent forms:

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

or

$$x \in \underline{R}X \text{ iff } [x]_R \subseteq X$$
$$x \in \overline{R}X \text{ iff } [x]_R \cap X \neq \emptyset.$$

The above three are interpreted as follows. The set $\underline{R}X$ is the set of all elements of U which can be *certainly* classified as elements of X in the knowledge R. The set $\overline{R}X$ is the set of elements of U which can be *possibly* classified as elements of X in R.

We define *R-positive region* ($POS_R(X)$), *R-negative region* ($NEG_R(X)$), and *R-borderline region* ($BN_R(X)$) of X as follows:

$$POS_R(X) = \underline{R}X$$
$$NEG_R(X) = U - \overline{R}X$$
$$BN_R(X) = \overline{R}X - \underline{R}X.$$

The positive region $POS_R(X)$ (or the lower approximation) of X is the collection of those objects which can be classified with full certainty as members of the set X, using knowledge R.

The negative region $NEG_R(X)$ is the collection of objects with which it can be determined without any ambiguity, employing knowledge R, that they do not belong to the set X, that is, they belong to the complement of X.

The borderline region $BN_R(X)$ is the set of elements which cannot be classified either to X or to $-X$ in R. It is the undecidable area of the universe, i.e. none of the objects belonging to the boundary can be classified with certainty into X or $-X$ as far as R is concerned.

Now, we list basic formal results. Their proofs may be found in Pawlak [1]. Proposition 2.1 is obvious.

Proposition 2.1 *The following hold:*

(1) X *is* R*-definable iff* $\underline{R}X = \overline{R}X$
(2) X *is rough with respect to* R *iff* $\underline{R}X \neq \overline{R}X$.

Proposition 2.2 shows the basic properties of approximations:

Proposition 2.2 *The* R*-lower and* R*-upper approximations satisfy the following properties:*

(1) $\underline{R}X \subseteq X \subseteq \overline{R}X$
(2) $\underline{R}\emptyset = \overline{R}\emptyset = \emptyset, \underline{R}U = \overline{R}U = U$
(3) $\overline{R}(X \cup Y) = \overline{R}X \cup \overline{R}Y$
(4) $\underline{R}(X \cap Y) = \underline{R}X \cap \underline{R}Y$
(5) $X \subseteq Y$ *implies* $\underline{R}X \subseteq \underline{R}Y$
(6) $X \subseteq Y$ *implies* $\overline{R}X \subseteq \overline{R}Y$
(7) $\underline{R}(X \cup Y) \supseteq \underline{R}X \cup \underline{R}Y$
(8) $\overline{R}(X \cap Y) \subseteq \overline{R}X \cap \overline{R}Y$
(9) $\underline{R}(-X) = -\overline{R}X$
(10) $\overline{R}(-X) = -\underline{R}X$
(11) $\underline{R}\underline{R}X = \overline{R}\underline{R}X = \underline{R}X$
(12) $\overline{R}\overline{R}X = \underline{R}\overline{R}X = \overline{R}X$

The concept of approximations of sets can be also applied to that of membership relation. In rough set theory, since the definition of a set is associated with knowledge about the set, a membership relation must be related to the knowledge.

Then, we can define two membership relations $\underline{\in}_R$ and $\overline{\in}_R$. $x \underline{\in}_R X$ reads "x *surely belongs* to X" and $\overline{\in}_R$ reads "x *possibly belongs* to X". $\underline{\in}_R$ and $\overline{\in}_R$ are called the *R-lower membership* and *R-upper membership*, respectively.

Proposition 2.3 states the basic properties of membership relations:

Proposition 2.3 *The R-lower and R-upper membership relations satisfy the following properties:*

(1) $x \underline{\in}_R X$ *imply* $x \in X$ *implies* $x \overline{\in}_R X$
(2) $X \subseteq Y$ *implies* $(x \underline{\in}_R X$ *implies* $x \underline{\in}_R Y$ *and* $x \overline{\in}_R X$ *implies* $x \overline{\in}_R Y)$
(3) $x \overline{\in}_R (X \cup Y)$ *iff* $x \overline{\in}_R X$ *or* $x \overline{\in}_R Y$
(4) $x \underline{\in}_R (X \cap Y)$ *iff* $x \underline{\in}_R X$ *and* $x \underline{\in}_R Y$
(5) $x \underline{\in}_R X$ *or* $x \underline{\in}_R Y$ *implies* $x \underline{\in}_R (X \cup Y)$
(6) $x \overline{\in}_R (X \cap Y)$ *implies* $x \overline{\in}_R X$ *and* $x \overline{\in}_R Y$
(7) $x \underline{\in}_R (-X)$ *iff non* $x \overline{\in}_R X$
(8) $x \overline{\in}_R (-X)$ *iff non* $x \underline{\in}_R X$

Approximate (rough) equality is the concept of equality in rough set theory. Three kinds of approximate equality can be introduced. Let $K = (U, \mathbf{R})$ be a knowledge base, $X, Y \subseteq U$ and $R \in IND(K)$.

(1) Sets X and Y are *bottom* R-equal ($X \bar{\sim}_R Y$) if $\underline{R}X = \underline{R}Y$
(2) Sets X and Y are *top* R-equal ($X \simeq_R Y$) if $\overline{R}X = \overline{R}Y$
(3) Sets X and Y are R-equal ($X \approx_R Y$) if $X \bar{\sim}_R Y$ and $X \simeq_R Y$

These equalities are equivalence relations for any indiscernibility relation R. They are interpreted as follows: $X \bar{\sim}_R Y$ means that positive example of the sets X and Y are the same, $(X \simeq_R Y)$ means that negative example of the sets X and Y are the same, and $(X \approx_R Y)$ means that both positive and negative examples of X and Y are the same.

These equalities satisfy the following proposition (we omit subscript R for simplicity):

Proposition 2.4 *For any equivalence relation, we have the following properties:*

(1) $X \bar{\sim} Y$ *iff* $X \cap X \bar{\sim} Y$ *and* $X \cap Y \bar{\sim} Y$
(2) $X \simeq Y$ *iff* $X \cup Y \simeq X$ *and* $X \cup Y \simeq Y$
(3) *If* $X \simeq X'$ *and* $Y \simeq Y'$*, then* $X \cup Y \simeq X' \cup Y'$
(4) *If* $X \bar{\sim} X'$ *and* $Y \bar{\sim} Y'$*, then* $X \cap Y \bar{\sim} X' \cap Y'$
(5) *If* $X \simeq Y$*, then* $X \cup -Y \simeq U$
(6) *If* $X \bar{\sim} Y$*, then* $X \cap -Y \bar{\sim} \emptyset$
(7) *If* $X \subseteq Y$ *and* $Y \simeq \emptyset$*, then* $X \simeq \emptyset$
(8) *If* $X \subseteq Y$ *and* $Y \simeq U$*, then* $X \simeq U$

(9) $X \simeq Y$ *iff* $-X \overset{-}{\sim} -Y$
(10) *If* $X \overset{-}{\sim} \emptyset$ *or* $Y \overset{-}{\sim} \emptyset$, *then* $X \cap Y \overset{-}{\sim} \emptyset$
(11) *If* $X \simeq U$ *or* $Y \simeq U$, *then* $X \cup Y \simeq U$.

The following proposition shows that lower and upper approximations of sets can be expressed by rough equalities:

Proposition 2.5 *For any equivalence relation R:*

(1) $\underline{R}X$ *is the intersection of all* $Y \subseteq U$ *such that* $X \overset{-}{\sim}_R Y$
(2) $\overline{R}X$ *is the union of all* $Y \subseteq U$ *such that* $X \simeq_R Y$.

Similarly, we can define rough inclusion of sets. It is possible to define three kinds of rough inclusions.

Let $X = (U, \mathbf{R})$ be a knowledge base, $X, Y \subseteq U$, and $R \in IND(K)$. Then, we have:

(1) Set X is *bottom* R*-included* in Y ($X \overset{\subset}{\sim}_R Y$) iff $\underline{R}X \subseteq \underline{R}Y$
(2) Set X is *top* R*-included* in Y ($X \overset{\sim}{\subset}_R Y$) iff $\overline{R}X \subseteq \overline{R}Y$
(3) Set X is R*-included* in Y ($X \overset{\sim}{\overset{\subset}{\sim}}_R Y$) iff $X \overset{\sim}{\subset}_R Y$ and $X \overset{\subset}{\sim}_R Y$.

Note that $\overset{\subset}{\sim}_R$, $\overset{\sim}{\subseteq}_R$ and $\overset{\sim}{\overset{\subset}{\sim}}_R$ are quasi ordering relations. They are called the lower, upper and rough inclusion relation, respectively. Observe that rough inclusion of sets does not imply the inclusion of sets.

The following proposition shows the properties of rough inclusion:

Proposition 2.6 *Rough inclusion satisfies the following:*

(1) *If* $X \subseteq Y$, *then* $X \overset{\subset}{\sim} Y$, $X \overset{\sim}{\subset} Y$ *and* $X \overset{\sim}{\overset{\subset}{\sim}} Y$
(2) *If* $X \overset{\subset}{\sim} Y$ *and* $Y \overset{\subset}{\sim} X$, *then* $X \overset{-}{\sim} Y$
(3) *If* $X \overset{\sim}{\subset} Y$ *and* $Y \overset{\sim}{\subset} X$, *then* $X \simeq Y$
(4) *If* $X \overset{\sim}{\overset{\subset}{\sim}} Y$ *and* $Y \overset{\sim}{\overset{\subset}{\sim}} X$, *then* $X \approx Y$
(5) *If* $X \overset{\sim}{\subset} Y$ *iff* $X \cup Y \simeq Y$
(6) *If* $X \overset{\subset}{\sim} Y$ *iff* $X \cap Y \overset{-}{\sim} Y$
(7) *If* $X \subseteq Y$, $X \overset{-}{\sim} X'$ *and* $Y \overset{-}{\sim} Y'$, *then* $X' \overset{\subset}{\sim} Y'$
(8) *If* $X \subseteq Y$, $X \simeq X'$ *and* $Y \simeq Y'$, *then* $X' \overset{\sim}{\subset} Y'$
(9) *If* $X \subseteq Y$, $X \approx X'$ *and* $Y \approx Y'$, *then* $X' \overset{\sim}{\overset{\subset}{\sim}} Y'$
(10) *If* $X' \overset{\sim}{\subset} X$ *and* $Y' \overset{\sim}{\subset} Y$, *then* $X' \cup Y' \overset{\sim}{\subset} X \cup Y$
(11) $X' \overset{\subset}{\sim} X$ *and* $Y' \overset{\subset}{\sim} Y$, *then* $X' \cap Y' \overset{\subset}{\sim} X \cap Y$
(12) $X \cap Y \overset{\subset}{\sim} Y \overset{\sim}{\subset} X \cup Y$
(13) *If* $X \overset{\subset}{\sim} Y$ *and* $X \overset{-}{\sim} Z$, *then* $Z \overset{\subset}{\sim} Y$

(14) If $X \overset{\sim}{\subset} Y$ *and* $X \simeq Z$*, then* $Z \overset{\sim}{\subset} Y$

(15) If $X \overset{\sim}{\overset{\subset}{\sim}} Y$ *and* $X \approx Z$*, then* $Z \overset{\sim}{\overset{\subset}{\sim}} Y$

The above properties are not valid if we replace $\overline{\sim}$ by $\simeq$ (or conversely). If R is an equivalence relation, then all three inclusions reduce to ordinary inclusion.

2.2 Variable Precision Rough Set Models

Ziarko generalized Pawlak's original rough set model in Ziarko [2], which is called the *variable precision rough set model* (VPRS model) to overcome the inability to model uncertain information, and is directly derived from the original model without any additional assumptions.

As the limitations of Pawlak's rough set model, Ziarko discussed two points. One is that it cannot provide a classification with a controlled degree of uncertainty. Some level of uncertainty in the classification process gives a deeper or better understanding for data analysis.

The other is that the original model has the assumption that the universe U of data objects is known. Therefore, all conclusions derived from the model are applicable only to this set of objects. It is useful to introduce uncertain hypotheses about properties of a larger universe.

Ziarko's extended rough set model generalizes the standard set inclusion relation, capable of allowing for some degree of misclassification in the largely correct classification.

Let X and Y be non-empty subsets of a finite universe U. X is included in Y, denoted $Y \supseteq X$, if for all $e \in X$ implies $e \in Y$. Here, we introduce the measure $c(X, Y)$ of the relative degree of misclassification of the set X with respect to set Y defined as:

$$c(X, Y) = 1 - \text{card}(X \cap Y)/\text{card}(X) \text{ if } \text{card}(X) > 0 \text{ or}$$
$$c(X, Y) = 0 \text{ if } \text{card}(X) = 0$$

where card denotes set cardinality.

The quantity $c(X, Y)$ will be referred to as the relative classification error. The actual number of misclassification is given by the product $c(X, Y) * \text{card}(X)$ which is referred to as an absolute classification error.

We can define the inclusion relationship between X and Y without explicitly using a general quantifier:

$$X \subseteq Y \text{ iff } c(X, Y) = 0$$

The *majority* requirement implies that more than 50% of X elements should be in common with Y. The *specified majority* requirement imposes an additional requirement. The number of elements of X in common with Y should be above 50% and not below a certain limit, e.g. 85%.

According to the specified majority requirement, the admissible classification error β must be within the range $0 \leq \beta < 0.5$. Then, we can define the majority inclusion relation based on this assumption.

$$X \overset{\beta}{\subseteq} Y \text{ iff } c(X, Y) \leq \beta$$

The above definition covers the whole family of β-majority relation. However, the majority inclusion relation does not have the transitivity relation.

The following two propositions indicate some useful properties of the majority inclusion relation:

Proposition 2.7 *If $A \cap B = \emptyset$ and $B \overset{\beta}{\supseteq} X$, then it is not true that $A \overset{\beta}{\supseteq} X$.*

Proposition 2.8 *If $\beta_1 < \beta_2$, then $Y \overset{\beta_1}{\supseteq} X$ implies $Y \overset{\beta_2}{\supseteq} X$.*

For the VPRS-model, we define the approximation space as a pair $A = (U, R)$, where U is a non-empty finite universe and R is the equivalence relation on U. The equivalence relation R, referred to as an indiscernibility relation, corresponds to a partitioning of the universe U into a collection of equivalence classes or elementary set $R^* = \{E_1, E_2, ..., E_n\}$.

Using a majority inclusion relation instead of the inclusion relation, we can obtain generalized notions of β-lower approximation (or β-positive region $\mathrm{POSR}_\beta(X)$) of the set $U \supseteq X$:

$$\underline{R}_\beta X = \bigcup\{E \in R^* : X \overset{\beta}{\supseteq} E\} \text{ or, equivalently,}$$

$$\underline{R}_\beta X = \bigcup\{E \in R^* : c(E, X) \leq \beta\}$$

The β-upper approximation of the set $U \supseteq X$ can be also defined as follows:

$$\overline{R}_\beta X = \bigcup\{E \in R^* : c(E, X) < 1 - \beta\}$$

The β-boundary region of a set is given by

$$\mathrm{BNR}_\beta X = \bigcup\{E \in R^* : \beta < c(E, X) < 1 - \beta\}.$$

The β-negative region of X is defined as a complement of the β-upper approximation:

$$\mathrm{NEGR}_\beta X = \bigcup\{E \in R^* : c(E, X) \geq 1 - \beta\}.$$

The lower approximation of the set X can be interpreted as the collection of all those elements of U which can be classified into X with the classification error not greater than β.

The β-negative region of X is the collection of all those elements of U which can be classified into the complement of X, with the classification error not greater than β. The latter interpretation follows from Proposition 2.9:

Proposition 2.9 *For every $X \subseteq Y$, the following relationship is satisfied:*

$$\mathrm{POSR}_{\beta}(-X) = \mathrm{NEGR}_{\beta} X.$$

The β-boundary region of X consists of all those elements of U which cannot be classified either into X or into $-X$ with the classification error not greater than β.

Notice here that the law of excluded middle, i.e. $p \vee \neg p$, where $\neg p$ is the negation of p, holds in general for imprecisely specified sets.

Finally, the β-upper approximation $\overline{R}_{\beta} X$ of X includes all those elements of U which cannot be classified into $-X$ with the error not greater than β. If $\beta = 0$ then the original rough set model is a special case of VPRS-model, as the following proposition states:

Proposition 2.10 *Let X be an arbitrary subset of the universe U:*

(1) $\underline{R}_0 X = \underline{R} X$, *where* $\underline{R} X$ *is a lower approximation defined as* $\underline{R} X = \bigcup\{E \in R^* : X \supseteq E\}$
(2) $\overline{R}_0 X = \overline{R} X$, *where* $\overline{R} X$ *is an upper approximation defined as* $\overline{R} X = \bigcup\{E \in R^* : E \cap X \neq \emptyset\}$
(3) $\mathrm{BNR}_0 X = \mathrm{BN}_R X$, *where* $\mathrm{BN}_R X$ *is the set* X *boundary region defined as* $\mathrm{BN}_R X = \overline{R} X - \underline{R} X$
(4) $\mathrm{NEGR}_0 X = \mathrm{NEG}_R X$, *where* $\mathrm{NEG}_R X$ *is the set* X *negative region defined as* $\mathrm{NEG}_R X = U - \overline{R} X$

In addition, we have the following proposition:

Proposition 2.11 *If $0 \leq \beta < 0.5$ then the properties listed in Proposition 2.10 and the following are also satisfied:*

$$\underline{R}_{\beta} X \supseteq \underline{R} X,$$
$$\overline{R} X \supseteq \overline{R}_{\beta} X,$$
$$\mathrm{BN}_R X \supseteq \mathrm{BNR}_{\beta} X,$$
$$\mathrm{NEGR}_{\beta} X \supseteq \mathrm{NEG}_R X.$$

Intuitively, with the decrease of the classification error β the size of the positive and negative regions of X will shrink, whereas the size of the boundary region will grow.

With the reduction of β fewer elementary sets will satisfy the criterion for inclusion in β-positive or βnegative regions. Thus, the size of the boundary will increase.

The reverse process can be done with the increase of β.

Proposition 2.12 *With the β approaching the limit 0.5, i.e., $\beta \to 0.5$, we obtain the following:*

$$
\begin{aligned}
&\underline{R}_\beta X \to \underline{R}_{0.5} X = \bigcup\{E \in R^* : c(E, X) < 0.5\},\\
&\overline{R}_\beta X \to \overline{R}_{0.5} X = \bigcup\{E \in R^* : c(E, X) \leq 0.5\},\\
&\mathrm{BNR}_\beta X \to \mathrm{BNR}_{0.5} X = \bigcup\{E \in R^* : c(E, X) = 0.5\},\\
&\mathrm{NEGR}_\beta X \to \mathrm{NEGR}_{0.5} X = \bigcup\{E \in R^* : c(E, X) > 0.5\}.
\end{aligned}
$$

The set $\mathrm{BNR}_{0.5} X$ is called an *absolute boundary* of X because it is included in every other boundary region of X.

The following Proposition 2.13 summarizes the primary relationships between set X discernibility regions computed on 0.5 accuracy level and higher levels.

Proposition 2.13 *For boundary regions of X, the following hold:*

$$
\begin{aligned}
&\mathrm{BRN}_{0.5} X = \bigcap_\beta \mathrm{BNR}_\beta X,\\
&\overline{R}_{0.5} X = \bigcap_\beta \overline{R}_\beta X,\\
&\underline{R}_{0.5} X = \bigcup_\beta \underline{R}_\beta X,\\
&\mathrm{NEGR}_{0.5} X = \bigcup_\beta \mathrm{NEGR}_\beta X.
\end{aligned}
$$

The absolute boundary is very “narrow”, consisting only of those sets which have 50/50 aplite of elements among set X interior and its exterior. All other elementary sets are classified either into positive region $\underline{R}_{0.5} X$ or the negative region $\mathrm{NEGR}_{0.5} X$.

We turn to the measure of approximation. To express the degree with which a set X can be approximately characterized by means of elementary sets of the approximation space $A = (U, R)$, we will generalize the accuracy measure introduced in Pawlak [3].

The β-accuracy for $0 \leq \beta < 0.5$ is defined as

$$\alpha(R, \beta, X) = \mathrm{card}(\underline{R}_\beta X)/\mathrm{card}(\overline{R}_\beta X).$$

The β-accuracy represents the imprecision of the approximate characterization of the set X relative to assumed classification error β.

Note that with the increase of β the cardinality of the β-upper approximation will tend downward and the size of the β-lower approximation will tend upward which leads to the conclusion that is consistent with intuition that relative accuracy may increase at the expense of a higher classification error.

The notion of discernibility of set boundaries is relative. If a large classification error is allowed then the set X can be highly discernable within assumed classification limits. When smaller values of the classification tolerance are assumed in may become more difficult to discern positive and negative regions of the set to meet the narrow tolerance limits.

The set X is said to be *β-discernable* if its β-boundary region is empty or, equivalently, if

$$\underline{R}_\beta X = \overline{R}_\beta X.$$

For the β-discernable sets the relative accuracy $\alpha(R, \beta, X)$ is equal to unity. The discernable status of a set change depending on the value of β. In general, the following properties hold:

Proposition 2.14 *If X is discernable on the classification error level* $0 \le \beta < 0.5$*, then X is also discernable at any level* $\beta_1 > \beta$*.*

Proposition 2.15 *If* $\overline{R}_{0.5}X \ne \underline{R}_{0.5}X$*, then X is not discernable on every classification error level* $0 \le \beta < 0.5$.

Proposition 2.16 emphasizes that a set with a non-empty absolute boundary can never be discerned. In general, one can easily demonstrate the following:

Proposition 2.16 *If X is not discernable on the classification error level* $0 \le \beta < 0.5$*, then X is also not discernible at any level* $\beta_1 < \beta$*.*

Any set X which is not discernable for every β is called indiscernible or absolutely rough. The set X is absolutely rough iff $\mathrm{BNR}_{0.5}X \ne \emptyset$. Any set which is not absolutely rough will be referred to as relatively rough or weakly discernable.

For each relatively rough set X, there exists such a classification error level β that X is discernable on this level.

Let $\mathrm{NDIS}(R, X) = \{0 \le \beta < 0.5 : \mathrm{BNR}_\beta(X) \ne \emptyset\}$.Then, $\mathrm{NDIS}(R, X)$ is a range of all those β values for which X is indiscernible.

The least value of classification error β which makes X discernable will be referred to as discernibility threshold. The value of the threshold is equal to the least upper bound $\zeta(R, X)$ of $\mathrm{NDIS}(X)$, i.e.,

$$\zeta(R, X) = \sup \mathrm{NDIS}(R, X).$$

Proposition 2.17 states a simple property which can be used to find the discernibility threshold of a weakly discernible set X:

Proposition 2.17 $\zeta(R, X) = \max(m_1, m_2)$*, where*

$$m_1 = 1 - \min\{c(E, X) : E \in R^* \text{ and } 0.5 < c(E, X)\},$$
$$m_2 = \max\{c(E, X) : E \in R^* \text{ and } c(E, X) < 0.5\}.$$

The discernibility threshold of the set X equals a minimal classification error β which can be allowed to make this set β-discernible.

We give some fundamental properties of β-approximations.

Proposition 2.18 *For every* $0 \le \beta < 0.5$*, the following hold:*

(1a) $X \overset{\beta}{\supseteq} \underline{R}_\beta X$
(1b) $\overline{R}_\beta X \supseteq \underline{R}_\beta X$
(2) $\underline{R}_\beta \emptyset = \overline{R}_\beta \emptyset = \emptyset$; $\underline{R}_\beta U = \overline{R}_\beta U = U$
(3) $\overline{R}_\beta (X \cup Y) \supseteq \overline{R}_\beta X \cup \overline{R}_\beta Y$

(4) $\underline{R}_\beta X \cap \underline{R}_\beta Y \supseteq \underline{R}_\beta(X \cap Y)$
(5) $\underline{R}_\beta(X \cup Y) \supseteq \underline{R}_\beta \cup \underline{R}_\beta Y$
(6) $\overline{R}_\beta X \cap \overline{R}_\beta Y \supseteq \overline{R}_\beta(X \cap Y)$
(7) $\underline{R}_\beta(-X) = -\overline{R}_\beta(X)$
(8) $\overline{R}_\beta(-X) = -\underline{R}_\beta(X)$

We finish the outline of variable precision rough set model, which can be regarded as a direct generalization of the original rough set model. Consult Ziarko [2] for more details. As we will be discussed later, it plays an important role in our approach to rough set based reasoning.

Shen and Wang [4] proposed the VPRS model over two universes using inclusion degree. They introduced the concepts of the reverse lower and upper approximation operators and studied their properties. They introduced the approximation operators with two parameters as a generalization of the VPRS-model over two universes.

2.3 Related Theories

There are many related theories which extend the original rough set theory in various aspects. Very interesting are theories which integrate both rough set theory and fuzzy set theory. In this section, we briefly review some of such theories.

Before describing related fuzzy-based rough set theories, we need to give a concise exposition of *fuzzy set theory*, although there are many possible descriptions in the literature.

Fuzzy set was proposed by Zadeh [5] to model fuzzy concepts, which cannot be formalized in classical set theory. Zadeh also developed a *theory of possibility* based on fuzzy set theory in Zadeh [6]. In fact, fuzzy set theory found many applications in various areas.

Let $\mathscr{U}$ be a set. Then, a fuzzy set is defined as follows:

Definition 2.1 (*Fuzzy set*) A *fuzzy set* of $\mathscr{U}$ is a function $u : \mathscr{U} \to [0, 1]$. $\mathscr{F}_\mathscr{U}$ will denote the set of all fuzzy sets of $\mathscr{U}$.

Several operations on fuzzy sets are defined as follows:

Definition 2.2 For all $u, v \in \mathscr{F}_\mathscr{U}$ and $x \in \mathscr{U}$, we put

$$(u \vee v)(x) = \sup\{u(x), v(x)\}$$
$$(u \wedge v)(x) = \inf\{u(x), v(x)\}$$
$$\overline{u}(x) = 1 - u(x)$$

Definition 2.3 Two fuzzy sets $u, v \in \mathscr{F}_\mathscr{U}$ are said to be *equal* iff for every $x \in \mathscr{U}$, $u(x) = v(x)$.

Definition 2.4 $\mathbf{1}_\mathscr{U}$ and $\mathbf{0}_\mathscr{U}$ are the fuzzy sets of $\mathscr{U}$ such that for all $x \in \mathscr{U}$, $\mathbf{1}_\mathscr{U} = 1$ and $\mathbf{0}_\mathscr{U} = 0$

It is easy to prove that $\langle \mathscr{F}_{\mathscr{U}}, \wedge, \vee \rangle$ is a complete lattice having infinite distributive property. Furthermore, $\langle \mathscr{F}_{\mathscr{U}}, \wedge, \vee, ^{-} \rangle$ constitutes an algebra, which in general is not Boolean (for details, see Negoita and Ralescu [7]).

Since both rough set theory and fuzzy set theory aim to formalize related notions, it is natural to integrate these two theories. In 1990, Dubois and Prade introduced *fuzzy rough sets* as a fuzzy generalization of rough sets.

They considered two types of generalizations. One is the upper and lower approximation of a fuzzy set, i.e., *rough fuzzy set*. The other provided an idea of turning the equivalence relation into a fuzzy similarity relation, yielding a *fuzzy rough set*.

Nakamura and Gao [8] also studied fuzzy rough sets and developed a logic for fuzzy data analysis. Their logic can be interpreted as a modal logic based on fuzzy relations. They related similarity relation on a set of objects to rough sets.

Quafafou [9] proposed *α-rough set theory* (α-RST) in 2000. In α-RST, all basic concepts of rough set theory are generalized. He described approximations of fuzzy concepts and their properties.

In addition, in α-RST, the notion of α-dependency, i.e., a set of attributes which depends on another with a given degree in [0, 1]., is introduced. It can be seen as a partial dependency. Note that α-RST has a feature of the ability of the control of the universe partitioning and the approximation of concepts.

Cornelis et al. [10] proposed *intuitionistic fuzzy rough sets* to describe incomplete aspects of knowledge based on *intuitionistic fuzzy sets* due to Atanassov [11] in 2003. Their approach adopted the idea that fuzzy rough sets should be intuitionistic.[1]

These works enhance the power of rough set theory by introducing fuzzy concepts in various ways. Fuzzy rough sets are more useful than the original rough sets, and they can be applied to more complicated problems.

2.4 Formal Concept Analysis

As a different area, *formal concept analysis* (FCA) has been developed; see Ganter and Wille [12]. It is based on *concept lattice* to model relations of concept in a precise way. Obviously, rough set theory and formal concept analysis share similar idea. We here present the outline of formal concept analysis in some detail.

FCA uses the notion of a formal concept as a mathematical formulation of the notion of a concept in Port-Royal logic. According to Port-Royal, a concept is determined by a collection of objects, called an *extent* which fall under the concept and a collection of attributes called an *intent* covered by the concepts. Concepts are ordered by a subconcept-superconcept relation which is based on inclusion relation on objects and attributes. We formally define these notions used in FCA.

[1]By intuitionistic, it means that the law of excluded middle fails. However, this does not always mean rough sets founded on the so-called intuitionistic logic.

A *formal context* is a triplet $\langle X, Y, I\rangle$, where X and Y are non-empty set, and I is a binary relation, i.e., $I \subseteq X \times Y$. Elements x from X are called *objects*, elements y from Y are called *attributes*, $\langle x, y\rangle \in I$ indicates that x has attribute y.

For a given cross-table with n rows and m columns, a corresponding formal context $\langle X, Y, I\rangle$ consists of a set $X = \{x_1, ..., x_n\}$, a set $Y = \{y_1, ..., y_m\}$, and a relation I defined by $\langle x_i, y_j\rangle \in I$ iff the table entry corresponding to row i and column j contains $\times$.

Concept-forming operators are defined for every formal context. For a formal context $\langle X, Y, I\rangle$, operators $^{\uparrow} : 2^X \to 2^Y$ and $^{\downarrow} : 2^Y \to 2^X$ are defined for every $A \subseteq X$ and $B \subseteq Y$ by

$$A^{\uparrow} = \{y \in Y \mid for\ each\ x \in A : \langle x, y\rangle \in I\}$$
$$B^{\downarrow} = \{x \in X \mid for\ each\ y \in B : \langle x, y\rangle \in I\}$$

Formal concepts are particular clusters in cross-tables, defined by means of attribute sharing. A *formal concept* in $\langle X, Y, I\rangle$ is a pair $\langle A, B\rangle$ of $A \subseteq X$ and $B \subseteq Y$ such that $A^{\uparrow} = B$ and $B^{\downarrow} = A$.

It is noticed that $\langle A, B\rangle$ is a formal concept iff A contains just objects sharing all attributes from B and B contains just attributes shared by all objects from A. Thus, mathematically $\langle A, B\rangle$ is a formal concept iff $\langle A, B\rangle$ is a fixpoint of the pair $\langle^{\uparrow},^{\downarrow}\rangle$ of the concept-forming operators.

Consider the following table:

I	y_1	y_2	y_3	y_4
x_1	×	×	×	×
x_2	×		×	×
x_3		×	×	×
x_4		×	×	×
x_5	×			

Here, formal concept

$$\langle A_1, B_1\rangle = \langle\{x_1, x_2, x_3, x_4\}, \{y_3, y_4\}\rangle$$

because

$$\{x_1, x_2, x_3, x_4\}^{\uparrow} = \{y_3, y_4\}$$
$$\{y_3, y_4\}^{\downarrow} = \{x_1, x_2, x_3, x_4\}$$

Here, the following relationships hold:

$$\{x_2\}^{\uparrow} = \{y_1, y_3, y_4\}, \{x_2, x_3\}^{\uparrow} = \{y_3, y_4\}$$
$$\{x_1, x_4, x_5\}^{\uparrow} = \emptyset$$
$$X^{\uparrow} = \emptyset, \emptyset^{\uparrow} = Y$$
$$\{y_1\}^{\downarrow} = \{x_1, x_2, x_5\}, \{y_1, y_2\}^{\downarrow} = \{x_1\}$$
$$\{y_2, y_3\}^{\downarrow} = \{x_1, x_3, x_4\}, \{y_2, y_3, y_4\}^{\downarrow} = \{x_1, x_3, x_4\}$$
$$\emptyset^{\downarrow} = X, Y^{\downarrow} = \{x_1\}$$

Concepts are naturally ordered by a subconcept-superconcept relation. The subconcept-superconcept relation, denoted $\leq$, is based on inclusion relation on objects and attributes. For formal concepts $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$ of $\langle A, Y, I\rangle$, $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$ iff $A_1 \subseteq A_2$ (iff $B_2 \subseteq B_1$).

In the above example, the following hold:

$$\langle A_1, B_1\rangle = \{\{x_1, x_2, x_3, x_4\}, \{y_3, y_4\}\}$$
$$\langle A_2, B_2\rangle = \{\{x_1, x_3, x_4\}, \{y_2, y_3, y_4\}\}$$
$$\langle A_3, B_3\rangle = \{\{x_1, x_2\}, \{y_1, y_3, y_4\}\}$$
$$\langle A_4, B_4\rangle = \langle\{x_1, x_2, x_5\}, \{y_1\}\rangle$$
$$\langle A_3, B_3\rangle \leq \langle A_1, B_1\rangle$$
$$\langle A_3, B_3\rangle \leq \langle A_4, B_4\rangle$$
$$\langle A_2, B_2\rangle \leq \langle A_1, B_1\rangle$$
$$\langle A_1, B_1\rangle \| \langle A_4, B_4\rangle \text{ (incomparable)}$$
$$\langle A_2, B_2\rangle \| \langle A_4, B_4\rangle$$
$$\langle A_3, B_3\rangle \| \langle A_2, B_2\rangle$$

We denote by $\mathscr{B}(X, Y, I)$ the collection of all formal concepts of $\langle X, Y, I\rangle$, i.e.,

$$\mathscr{B}(X, Y, I) = \{\langle A, B\rangle \in 2^X \times 2^X \mid A^{\uparrow} = B, B^{\downarrow} = A\}.$$

$\mathscr{B}(X, Y, I)$ equipped with the subconcept-superconcept ordering $\leq$ is called a *concept lattice* of $\langle X, Y, I\rangle$. $\mathscr{B}(X, Y, I)$ represents all clusters which are hidden in data $\langle X, Y, I\rangle$. We can see that $\langle \mathscr{B}(X, Y, I), \leq\rangle$ is a lattice.

Extents and intents of concepts are defined as follows:

$$\mathrm{Ext}(X, Y, I) = \{A \in 2^X \mid \langle A, B\rangle \in \mathscr{B}(X, Y, I)\ for\ some\ B\} \text{ (extent of concepts)}$$
$$\mathrm{Int}(X, Y, I) = \{A \in 2^Y \mid \langle A, B\rangle \in \mathscr{B}(X, Y, I)\ for\ some\ A\} \text{ (intent of concepts)}$$

Formal concepts can be also defined as maximal rectangles in the cross-table: A rectangle in $\langle X, Y, I\rangle$ is a pair $\langle A, B\rangle$ such that $A \times B \subseteq I$, i.e., for each $x \in A$ and $y \in B$ we have $\langle x, y\rangle \in I$. For rectangles $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$, put $\langle A_1, B_1\rangle \sqsubseteq \langle A_2, B_2\rangle$ iff $A_1 \subseteq A_2$ and $B_1 \subseteq B_2$.

We can prove that $\langle A, B\rangle$ is a formal concept of $\langle X, Y, I\rangle$ iff $\langle A, B\rangle$ is a maximal rectangle in $\langle X, Y, I\rangle$. Consider the following table.

I	y_1	y_2	y_3	y_4
x_1	×	×	×	×
x_2	×		×	×
x_3		×	×	×
x_4		×	×	×
x_5	×			

In this table, $\langle \{x_1, x_2, x_3\}, \{y_3, y_4\} \rangle$ is a rectangle which is not maximal with respect to $\sqsubseteq$. $\langle \{x_1, x_2, x_3, x_4\}, \{y_3, y_4\} \rangle$ is a rectangle which is maximal with respect to $\sqsubseteq$. The notion of rectangle can server as a basis for geometrical reasoning in formal concept analysis.

There are two basic mathematical structures behind formal concept analysis, i.e., Galois connections (cf. Ore [13]) and closure operators. A *Galois connection* between sets X and Y is a pair $\langle f, g \rangle$ of $f : 2^X \to 2^Y$ and $g : 2^Y \to 2^X$ satisfying $A, A_1, A_2, B, B_1, B_2 \subseteq Y$:

$$A_1 \subseteq A_2 \;\Rightarrow\; f(A_2) \subseteq f(A_1)$$
$$B_1 \subseteq B_2 \;\Rightarrow\; g(B_2) \subseteq f(B_1)$$
$$A \subseteq g(f(A))$$
$$B \subseteq f(g(B)).$$

For a Galois connection $\langle f, g \rangle$ between sets X and Y, the set:

$$\mathrm{fix}(\langle f, g \rangle) = \{\langle A, B \rangle \in 2^X \times 2^X \mid f(A) = B, g(B) = A\}$$

is called a set of *fixpoint* of $\langle f, g \rangle$.

Here, we show a basic property of concept-forming operators. That is, for a formal context $\langle X, Y, I \rangle$, the pair $\langle ^{\uparrow_I}, ^{\downarrow_I} \rangle$ of operators induced by $\langle X, Y, I \rangle$ is a Galois connection between X and Y.

As consequence of the property, it is shown that for a Galois connection $\langle f, g \rangle$ between X and Y, $f(A) = f(g(f(A)))$ and $g(B) = g(f(g(B)))$ for any $A \subseteq X$ and $B \subseteq Y$.

Closure operators result from the concept-forming operators by their composition. If $\langle f, g \rangle$ is a Galois connection between X and Y, then $C_X = g \circ f$ is a closure operator on X and $C_Y = f \circ g$ is a closure operator on Y.

We can show that extents and intents are just the images under the concept-forming operators as follows:

$$\mathrm{Ext}(X, Y, I) = \{B^{\downarrow} \mid B \subseteq Y\}$$
$$\mathrm{Int}(X, Y, I) = \{A^{\uparrow} \mid A \subseteq X\}.$$

The following relationships hold for any formal context $\langle X, Y, I \rangle$:

$$\mathrm{Ext}(X, Y, I) = \mathrm{fix}(^{\uparrow\downarrow})$$
$$\mathrm{Int}(X, Y, I) = \mathrm{fix}(^{\downarrow\uparrow})$$
$$\mathscr{B}(X, Y, I) = \{\langle A, A^{\uparrow} \rangle \mid A \in \mathrm{Ext}(X, Y, I)\}$$
$$\mathscr{B}(X, Y, I) = \{\langle B^{\downarrow}, B \rangle \mid B \in \mathrm{Int}(X, Y, I)\}$$

The above definition of Galois connection can be simplified by the following simplified form. $\langle f, g \rangle$ is a Galois connection between X and Y iff for every $A \subseteq X$ and $B \subseteq Y$:

$$A \subseteq g(B) \text{ iff } B \subseteq f(A).$$

Galois connections with respect to union and intersection satisfy the following properties: Let $\langle f, g\rangle$ be a Galois connection between X and Y. For $A_j \subseteq X$, $j \in J$ and $B_j \subseteq Y$, $j \in J$, we have:

$$f(\bigcup_{j\in J} A_j) = \bigcap_{j\in J} f(A_j)$$
$$g(\bigcup_{j\in J} B_j) = \bigcap_{j\in J} g(B_j)$$

Every pair of concept-forming operators forms a Galois connection, and every Galois connection is a concept-forming operator of a particular formal context.

Let $\langle f, g\rangle$ be a Galois connection between X and Y. Consider a formal context $\langle X, Y, I\rangle$ such that I is defined by

$$\langle x, y\rangle \in I \text{ iff } y \in f(\{x\}) \text{ or equivalently, iff } x \in g(\{y\})$$

for each $x \in X$ and $y \in Y$. Then, $\langle {}^{\uparrow_I}, {}^{\downarrow_I}\rangle = \langle f, g\rangle$, i.e., $\langle {}^{\uparrow_I}, {}^{\downarrow_I}\rangle$ induced by $\langle X, Y, I\rangle$ coincide with $\langle f, g\rangle$.

We can establish representation result in the following form, i.e., $I \mapsto \langle {}^{\uparrow_I}, {}^{\downarrow_I}\rangle$ and $\langle {}^{\uparrow_I}, {}^{\downarrow_I}\rangle \mapsto I_{\langle\uparrow,\downarrow\rangle}$ are mutually inverse mappings between the set of all binary relations between X and Y and the set of all Galois connections between X and Y.

We can also see the duality relationships between extents and intents. For $\langle A_1, B_1\rangle, \langle A_2, B_2\rangle \in \mathscr{B}(X, Y, I)$, we have that $A_1 \subseteq A_2$ iff $B_2 \subseteq B_1$. Then, we have the following peoperties:

(1) $\langle \mathrm{Ext}(X, Y, I), \subseteq\rangle$ and $\langle \mathrm{Int}(X, Y, I), \subseteq\rangle$ are partially ordered sets.
(2) $\langle \mathrm{Ext}(X, Y, I), \subseteq\rangle$ and $\langle \mathrm{Int}(X, Y, I), \subseteq\rangle$ are dually isomorphic, i.e., there is a mapping $f : \mathrm{Ext}(X, Y, I) \to \mathrm{Int}(X, Y, I)$ satisfying $A_1 \subseteq A_2$ iff $f(A_2) \subseteq f(A_1)$
(3) $\langle \mathscr{B}(X, Y, I), \leq\rangle$ is isomorphic to $\langle \mathrm{Ext}(X, Y, I), \subseteq\rangle$
(4) $\langle \mathscr{B}(X, Y, I), \leq\rangle$ is dually isomorphic to $\langle \mathrm{Int}(X, Y, I), \subseteq\rangle$.

We can also state the property of fixpoints of closure operators. For a closure operator C on X, the partially ordered set $\langle \mathrm{fix}(C), \subseteq\rangle$ of fixpoints of C is a complete lattice with infima and suprema given by:

$$\bigwedge_{j\in J} A_j = C(\bigcap_{j\in J} A_j)$$
$$\bigvee_{j\in J} A_j = C(\bigcup_{j\in J} A_j)$$

The following is the main result of concept lattices due to Wille.

(1) $\mathscr{B}(X, Y, I)$ is a complete lattice with infima and suprema given by

$$\bigwedge_{j\in J}\langle A_j, B_j\rangle = \langle \bigcap_{j\in J} A_j, (\bigcup_{j\in J} B_j)^{\downarrow\uparrow}\rangle, \bigvee_{j\in J}\langle A_j, B_j\rangle = \langle (\bigcup_{j\in J} A_j)^{\uparrow\downarrow}, \bigcap_{j\in J} B_j\rangle.$$

(2) Moreover, an arbitrary complete lattice $\mathbf{V} = (V, \leq)$ is isomorphic to $\mathscr{B}(X, Y, I)$ iff there are mapping $\gamma : X \to V$, $\mu : Y \to X$ such that

(i) $\gamma(X)$ is $\bigvee$-dense in V, $\mu(Y)$ is $\bigwedge$-dense in V
(ii) $\gamma(x) \leq \mu(y)$ iff $\langle x, y\rangle \in I$

In formal concept analysis, we can clarify and reduce formal concepts by removing some of objects or attributes in a formal context. A formal context $\langle X, Y, I\rangle$ is called *clarified* if the corresponding table neither contain identical rows nor identical columns. Namely, if $\langle X, Y, I\rangle$ is clarified then:

$\{x_1\}^{\uparrow} = \{x_2\}^{\uparrow}$ implies $x_1 = x_2$ for every $x_1, x_2 \in X$,
$\{y_1\}^{\downarrow} = \{y_2\}^{\downarrow}$ implies $y_1 = y_2$ for every $y_1, y_2 \in Y$.

Clarification can be performed by removing identical rows and columns. If $\langle X_1, Y_1, I_1\rangle$ is a clarified context resulting from $\langle X_2, Y_2, I_2\rangle$ by clarification, then $\mathscr{B}(X_1, Y_1, I_1)$ is isomorphic to $\mathscr{B}(X_2, Y_2, I_2)$.

For a formal context $\langle X, Y, I\rangle$, an attribute $y \in Y$ is called *reducible* iff there is $Y' \subset Y$ with $y \notin Y'$ such that

$$\{y\}^{\downarrow} = \bigcap_{z \in Y'} \{z\}^{\downarrow}$$

i.e., the column corresponding to y is the intersection of columns corresponding to z's from Y'.

An object $x \in X$ is called *reducible* iff there is $X' \subset X$ with $x \notin X'$ such that

$$\{x\}^{\uparrow} = \bigcap_{z \in X'} \{z\}^{\uparrow}$$

i.e., the row corresponding to x is the intersection of columns corresponding to z's from X'.

Let $y \in Y$ be reducible in $\langle X, Y, I\rangle$. Then, $\mathscr{B}(X, Y - \{y\}, J)$ is isomorphic to $\mathscr{B}(X, Y, I)$, where $J = I \cap (X \times (Y - \{y\}))$ is the restriction of I to $X \times Y - \{y\}$, i.e., $\langle X, Y - \{y\}, J\rangle$ results by removing y from $\langle X, Y, I\rangle$.

$\langle X, Y, I\rangle$ is *row reducible* if no object $x \in X$ is reducible; it is *column reducible* if no attribute $y \in Y$ is reducible; it is *reduced* if it is both row reduced and column reduced.

Arrow relations can find which objects and attributes are reducible. For $\langle X, Y, I\rangle$, we define relations $\nearrow$, $\swarrow$, $\updownarrow$ between X and Y:

$x \swarrow y$ iff $\langle x, y\rangle \notin I$ and if $\{x\}^{\uparrow} \subset \{x_1\}^{\uparrow}$ then $\langle x_1, y\rangle \in I$
$x \nearrow y$ iff $\langle x, y\rangle \notin I$ and if $\{x\}^{\downarrow} \subset \{x_1\}^{\downarrow}$ then $\langle x_1, y\rangle \in I$
$x \updownarrow y$ iff $x \swarrow y$ and $x \nearrow y$.

Thus, if $\langle x, y\rangle \in I$, then none of the above three relations occurs. Consequently, the arrow relations can be entered in the table of $\langle X, Y, I\rangle$. There is the following connections between arrow relations and reducibility.

$\langle \{x\}^{\uparrow\downarrow}, \{x\}^{\uparrow}\rangle$ is $\bigvee$-irreducible iff there is $y \in Y$ such that $x \swarrow y$

$\langle \{y\}^{\downarrow}, \{y\}^{\downarrow\uparrow}\rangle$ is $\bigvee$-irreducible iff there is $x \in X$ such that $x \nearrow y$.

Formal concept analysis can also deal with *attribute implication* concerning dependencies of data. Let Y be a non-empty set of attributes.

An attribute implication over Y is an expression

$$A \Rightarrow B$$

where $A, B \subseteq Y$.

An attribute implication $A \Rightarrow B$ over Y is *true* (valid) in a set $M \subset Y$ iff $A \subseteq M$ implies $B \subseteq M$. We write $\|A \Rightarrow B\|_M = 1$ (0) if $A \Rightarrow B$ is true (false) in M.

Let M be a set of attributes of some object x, $\|A \Rightarrow B\|_M = 1$ says "if x has all attributes from A then x has all attributes from B", because "if x has all attributes from C" is equivalent to $C \subseteq M$.

It is possible to extend the validity of $A \Rightarrow B$ to collections $\mathscr{M}$ of M's (collections of subsets of attributes), i.e., define validity of $A \Rightarrow B$ in $\mathscr{M} \subseteq 2^Y$.

An attribute implication $A \Rightarrow B$ over Y is true (valid) in $\mathscr{M}$ if $A \Rightarrow B$ is true in each $M \in \mathscr{M}$. An attribute implication $A \Rightarrow B$ over Y is true (valid) in a table (formal context) $\langle X, Y, I\rangle$ iff $A \Rightarrow B$ is true in $\mathscr{M} = \{\{x\}^{\uparrow} \mid x \in X\}$.

We define semantic consequence (entailment). An attribute implication $A \Rightarrow B$ follows semantically from a theory T, denoted $T \models A \Rightarrow B$ iff $A \Rightarrow B$ is true in every model M of T.

The system for reasoning about attribute implications consists of the following deduction rules:

(Ax) infer $A \cup B \Rightarrow A$,

(Cut) from $A \Rightarrow B$ and $B \cup C \Rightarrow D$ infer $A \cup C \Rightarrow D$.

Note that the above deduction rules are due to Armstrong's work on functional dependencies in databases; see Armstrong [14].

A *proof* of $A \Rightarrow B$ from a set T of attribute implications is a sequence $A_1 \Rightarrow B_1, \ldots, A_n \Rightarrow B_n$ of attribute implications satisfying:

(1) $A_n \Rightarrow B_n$ is just $A \Rightarrow B$,
(2) for every $i = 1, 2, \ldots, n$:

either $A_i \Rightarrow B_i \in T$ (assumption)
or $A_i \Rightarrow B_i$ results by using (Ax) or (Cut) from preceding attribute implications $A_j \Rightarrow B_j$'s (deduction)

If we have a proof of $A \Rightarrow B$ from T, then we write $T \vdash A \Rightarrow B$. We have the following derivable rules:

(Ref) infer $A \Rightarrow A$,

(Wea) from $A \Rightarrow B$, infer $A \cup C \Rightarrow B$,

(Add) from $A \Rightarrow B$ and $A \Rightarrow C$, infer $A \Rightarrow B \cup C$,

(Pro) from $A \Rightarrow B \cup C$, infer $A \Rightarrow B$,

(Tra) from $A \Rightarrow B$ and $B \Rightarrow C$, infer $A \Rightarrow C$,

for every $A, B, C, D \subseteq Y$.

We can show that (Ax) and (Cut) are sound. It is also possible to prove soundness of above derived rules.

We can define two notions of consequence, i.e., *semantic consequence* and *syntactic consequence*:

Semantic: $T \models A \Rightarrow B (A \Rightarrow B$ semantically follows from $T)$

Syntactic: $T \vdash A \Rightarrow B (A \Rightarrow B$ syntactically follows from $T)$

Semantic closure of T is the set

$$sem(T) = \{A \Rightarrow B \mid T \models A \Rightarrow B\}$$

of all attribute implications which semantically follows from T.

Syntactic closure of T is the set

$$syn(T) = \{A \Rightarrow B \mid T \vdash A \Rightarrow B\}$$

of all attribute implications which syntactically follows from T.

T is *semantically closed* if $T = sem(T)$. T is *syntactically closed* if $T = syn(T)$. Note that $sem(T)$ is the least set of attribute implications which is semantically closed containinig T and that $syn(T)$ is the least set of attribute implications which is syntactically closed containinig T.

It can be proved that T is syntactically closed iff for any $A, B, C, D \subseteq Y$

(1) $A \cup B \Rightarrow B \in T$,
(2) if $A \Rightarrow B \in T$ and $B \cup C \Rightarrow D \in T$ implies $A \cup C \Rightarrow D \in T$.

Then, if T is semantically closed, then T is syntactically closed. It can also be proved that if T is syntactically closed, then T is semantically closed. Consequently, soundness and completeness follow:

$$T \vdash A \Rightarrow B \text{ iff } T \models A \Rightarrow B.$$

We turn to models of attribute implications. For a set T of attribute implications, denote

$$Mod(T) = \{M \subseteq Y \mid \|A \Rightarrow B\|_M = 1 \; for \; every \; A \Rightarrow B \in T\}$$

That is, $Mod(T)$ is the set of all models of T.

A closure system in a set of Y is any system $\mathscr{S}$ of subsets of Y which contains Y and is closed under arbitrary intersections. That is, $Y \in \mathscr{S}$ and $\bigcap \mathscr{R} \in \mathscr{S}$ for every $\mathscr{R} \subseteq \mathscr{S}$ (intersection of every subsystem $\mathscr{R}$ of $\mathscr{S}$ belongs to $\mathscr{S}$.

There is a one-to-one relationship between closure systems in Y and closure operators in Y. Namely, for a closure operator C in Y, $\mathscr{S}_C = \{A \in 2^X \mid A = C(A)\} = \text{fix}(C)$ is a closure system in Y.

Given a closure system in Y, we set

$$C_{\mathscr{S}}(A) = \bigcap \{B \in \mathscr{S} \mid A \subseteq B\}$$

for any $A \subseteq X$, $C_{\mathscr{S}}$ is a closure operator on Y. This is a one-to-one relationship, i.e., $C = C_{\mathscr{S}_C}$ and $\mathscr{S} = \mathscr{S}_{C_{\mathscr{S}}}$.

It is shown that for a set of T of attribute implications, $Mod(T)$ is a closure system in Y. Since Mod(T) is a closure system, we can consider the corresponding closure operator $C_{\text{Mod}(T)}$, i.e., the fixpoints of $C_{\text{Mod}(T)}$ are just models of T.

Therefore, for every $A \subseteq Y$, there exists the least model of $Mod(T)$ which contains A, namely such least models is just $C_{\text{Mod}(T)}(A)$.

We can test entailment via least models follows. For any $A \Rightarrow B$ and any T, we have:

$$T \models A \Rightarrow B \text{ iff } \|A \Rightarrow B\|_{C_{Mod(T)}(A)} = 1.$$

It follows that the deductive system for attribute implications is sound and complete. And it can serve as a basis for reasoning about dependencies.

As reviewed in this section, formal concept analysis offers an interesting tool for data analysis. It has a mathematical foundation based on concept lattice with reasoning mechanisms based on attribute implications. In addition, formal concept analysis can visualize data.

Because formal concept analysis uses the notion of table, some similarities with rough set theory may be found. In fact, it uses classical (two-valued) basis. However, it is not clear whether it is possible to modify it, relating to some non-classical logic.

2.5 Decision Logic

Pawlak developed *decision logic* (DL) for reasoning about knowledge. His main goal is reasoning about knowledge concerning reality. Knowledge is represented as a value-attribute table, called *knowledge representation system*.

There are several advantages to represent knowledge in tabular form. The data table can be interpreted differently, namely it can be formalized as a logical system. The idea leads to decision logic.

The language of DL consists of atomic formulas, which are attribute-value pairs, combined by logical connectives to form compound formulas. The alphabet of the language consists of:

1. A: the set of attribute constants
2. $V = \bigcup V_a$: the set of attribute constants $a \in A$
3. Set $\{\sim, \vee, \wedge, \rightarrow, \equiv\}$ of propositional connectives, called negation, disjunction, conjunction, implication and equivalence, respectively.

The set of formulas in DL-language is the least set satisfying the following conditions:

1. Expressions of the form (a, v), or in short a_v, called atomic formulas, are formulas of DL-language for any $a \in A$ and $v \in V_a$.
2. If ϕ and ψ are formulas of DL-language, then so are $\sim \phi$, $(\phi \vee \psi)$, $(\phi \wedge \psi)$, $(\phi \rightarrow \psi)$ and $(\phi \equiv \psi)$.

Formulas are used as descriptions of objects of the universe. In particular, atomic formula of the form (a, v) is interpreted as a description of all objects having value v for attribute a.

The semantics for DL is given by a model. For DL, the model is KR-system $S = (U, A)$, which describes the meaning of symbols of predicates (a, v) in U, and if we properly interpret formulas in the model, then each formula becomes a meaningful sentence, expressing properties of some objects.

An object $x \in U$ *satisfies* a formula ϕ in $S = (U, A)$, denoted $x \models_S \phi$ or in short $x \models \phi$, iff the following conditions are satisfied:

1. $x \models (a, v)$ iff $a(x) = v$
2. $x \models \sim \phi$ iff $x \not\models \phi$
3. $x \models \phi \vee \psi$ iff $x \models \phi$ or $x \models \psi$
4. $x \models \phi \wedge \psi$ iff $x \models \phi$ and $x \models \psi$

The following are clear from the above truth definition:

5. $x \models \phi \rightarrow \psi$ iff $x \models \sim \phi \vee \psi$
6. $x \models \phi \equiv \psi$ iff $x \models \phi \rightarrow \psi$ and $x \models \psi \rightarrow \phi$

If ϕ is a formula, then the set $|\phi|_S$ defined as follows:

$$|\phi|_s = \{x \in U \mid x \models_S \phi\}$$

will be called the *meaning* of the formula ϕ in S.

Proposition 2.19 *The meaning of arbitrary formulas satisfies the following:*

$|(a, v)|_S = \{x \in U \mid a(x) = v\}$
$|\sim \phi|_S = - |\phi|_S$
$|\phi \vee \psi|_S = |\phi|_S \cup |\psi|_S$
$|\phi \wedge \psi|_S = |\phi|_S \cap |\psi|_S$
$|\phi \rightarrow \psi|_S = - |\phi|_S \cup |\psi|_S$

$$|\phi \equiv \psi|_S = (|\phi|_S \cap |\psi|_S) \cup (-|\phi|_S \cap - |\psi|_S)$$

Thus, the meaning of the formula ϕ is the set of all objects having the property expressed by the formula ϕ, or the meaning of the formula ϕ is the description in the KR-language of the set objects $|\phi|$.

A formula ϕ is said to be *true* in a KR-system S, denoted $\models_S \phi$, iff $|\phi|_S = U$, i.e., the formula is satisfied by all objects of the universe in the system S. Formulas ϕ and ψ are equivalent in S iff $|\phi|_S = |\psi|_S$.

Proposition 2.20 *The following are the simple properties of the meaning of a formula.*

$\models_S \phi$ *iff* $|\phi| = U$
$\models_S \sim \phi$ *iff* $|\phi| = \emptyset$
$\phi \to \psi$ *iff* $|\psi| \subseteq |\psi|$
$\phi \equiv \psi$ *iff* $|\psi| = |\psi|$

The meaning of the formula depends on the knowledge we have about the universe, i.e., on the knowledge representation system. In particular, a formula may be true in one knowledge representation system, but false in another one.

However, there are formulas which are true independent of the actual values of attributes appearing them. But, they depend only on their formal structure.

Note that in order to find the meaning of such a formula, one need not be acquainted with the knowledge contained in any specific knowledge representation system because their meaning is determined by its formal structure only.

Hence, if we ask whether a certain fact is true in light of our actual knowledge, it is sufficient to use this knowledge in an appropriate way. For formulas which are true (or not) in every possible knowledge representation system, we do not need in any particular knowledge, but only suitable logical tools.

To deal with deduction in DL, we need suitable axioms and inference rules. Here, axioms will correspond closely to axioms of classical propositional logic, but some specific axioms for the specific properties of knowledge representation systems are also needed. The only inference rule will be *modus ponens*.

We will use the following abbreviations:

$$\phi \wedge \sim \phi =_{\text{def}} 0$$

$$\phi \vee \sim \phi =_{\text{def}} 1$$

Obviously, $\models 1$ and $\models \sim 0$. Thus, 0 and 1 can be assumed to denote *falsity* and *truth*, respectively.

Formula of the form:

$$(a_1, v_1) \wedge (a_2, v_2) \wedge \ldots \wedge (a_n, v_n)$$

where $v_{a_i} \in V_a$, $P = \{a_1, a_2, \ldots, a_n\}$ and $P \subseteq A$ is called a P*-basic formula* or in short P-formula. Atomic formulas is called A*-basic formula* or in short basic formula.

Let $P \subseteq A, \phi$ be a P-formula and $x \in U$. If $x \models \phi$ then ϕ is called the *P-description* of x in S. The set of all A-basic formulas satisfiable in the knowledge representation system $S = (U, A)$ is called the *basic knowledge* in S.

We write $\sum_S(P)$, or in short $\sum(P)$, to denote the disjunction of all P-formulas satisfied in S. If $P = A$ then $\sum(A)$ is called the *characteristic formula* of S.

The knowledge representation system can be represented by a data table. And its columns are labelled by attributes and its rows are labelled by objects. Thus, each row in the table is represented by a certain A-basic formula, and the whole table is represented by the set of all such formulas. In DL, instead of tables, we can use sentences to represent knowledge.

There are specific axioms of DL:

1. $(a, v) \wedge (a, u) \equiv 0$ for any $a \in A, u, v \in V$ and $v \neq u$
2. $\bigvee_{v \in V_a} (a, v) \equiv 1$ for every $a \in A$
3. $\sim (a, v) \equiv \bigvee_{a \in V_a, u \neq v}(a, u)$ for every $a \in A$

The axiom (1) states that each object can have exactly one value of each attribute.

The axiom (2) assumes that each attribute must take one of the values of its domain for every object in the system.

The axiom (3) allows us to eliminate negation in such a way that instead of saying that an object does not possess a given property we can say that it has one of the remaining properties.

Proposition 2.21 *The following holds for DL:*

$$\models_S \sum_S(P) \equiv 1 \textit{ for any } P \subseteq A.$$

Proposition 2.21 means that the knowledge contained in the knowledge representation system is the whole knowledge available at the present stage. and corresponds to the so-called *closed world assumption* (CWA).

We say that a formula ϕ is *derivable* from a set of formulas Ω, denoted $\Omega \vdash \phi$, iff it is derivable from axioms and formulas of Ω by finite application of *modus ponens*. Formula ϕ is a theorem of DL, denoted $\vdash \phi$, if it is derivable from the axioms only. A set of formulas Ω is *consistent* iff the formula $\phi \wedge \sim \phi$ is not derivable from Ω.

Note that the set of theorems of DL is identical with the set of theorems of classical propositional logic with specific axioms (1)–(3), in which negation can be eliminated.

Formulas in the *KR*-language can be represented in a special form called *normal form*, which is similar to that in classical propositional logic.

Let $P \subseteq A$ be a subset of attributes and let ϕ be a formula in *KR*-language. We say that ϕ is in a *P-normal form* in S, in short in P-normal form, iff either ϕ is 0 or ϕ is 1, or ϕ is a disjunction of non-empty P-basic formulas in S. (The formula ϕ is non-empty if $|\phi| \neq \emptyset$).

Table 2.1 *KR*-system 1

U	a	b	c
1	1	0	2
2	2	0	3
3	1	1	1
4	1	1	1
5	2	1	3
6	1	0	3

A-normal form will be referred to as *normal form*. The following is an important property in the DL-language.

Proposition 2.22 *Let ϕ be a formula in DL-language and let P contain all attributes occurring in ϕ. Moreover, (1)–(3) and the formula $\sum_S(A)$. Then, there is a formula ψ in the P-normal form such that $\phi \equiv \psi$.*

Here is the example from Pawlak [1]. Consider the following *KR*-system (Table 2.1).

The following $a_1b_0c_2$, $a_2b_0c_3$, $a_1b_1c_1$, $a_2b_1c_3$, $a_1b_0c_3$ are all basic formulas (basic knowledge) in the *KR*-system. For simplicity, we will omit the symbol of conjunction $\wedge$ in basic formulas.

The characteristic formula of the system is:

$$a_1b_0c_2 \vee a_2b_0c_3 \vee a_1b_1c_1 \vee a_2b_1c_3 \vee a_1b_0c_3$$

Here, we give the following meanings of some formulas in the system:

$$|a_1 \vee b_0c_2| = \{1, 3, 4, 6\}$$
$$|\sim (a_2b_1)| = \{1, 2, 3, 4, 6\}$$
$$|b_0 \to c_2| = \{1, 3, 4, 5\}$$
$$|a_2 \equiv b_0| = \{2, 3, 4\}$$

Below are given normal forms of formulas considerred in the above example for *KR*-system 1:

$$a_1 \vee b_0c_2 = a_1b_0c_2 \vee a_1b_1c_1 \vee a_1b_0c_3$$
$$\sim (a_2b_1) = a_1b_0c_2 \vee a_2b_0c_3 \vee a_1b_1c_1 \vee a_1b_0c_3$$
$$b_0 \to c_2 = a_1b_0c_2 \vee a_1b_1c_1 \vee a_2b_1c_3$$
$$a_2 \equiv b_0 = a_2b_0c_1 \vee a_2b_0c_2 \vee a_2b_0c_3 \vee a_1b_1c_1 \vee a_1b_1c_2 \vee a_1b_1c_3$$

Examples of formulas in $\{a, b\}$-normal form are:

$$\sim (a_2b_1) = a_1b_0 \vee a_2b_0 \vee a_1b_1 \vee a_1b_0$$
$$a_2 \equiv b_0 = a_2b_0 \vee a_1b_1$$

The following is an example of a formula in $\{b, c\}$-normal form:

$$b_0 \rightarrow c_2 = b_0c_2 \vee b_1c_1 \vee b_1c_3$$

Thus, in order to compute the normal form of a formula, we have to transform by using propositional logic and the specific axioms for a given *KR*-system.

Any implication $\phi \rightarrow \psi$ is called a *decision rule* in the *KR*-language. ϕ and ψ are referred to as the *predecessor* and *successor* of $\phi \rightarrow \psi$, respectively.

If a decision rule $\phi \rightarrow \psi$ is true in S, we say that the decision rule is *consistent* in S; otherwise the decision rule is *inconsistent* in S.

If $\phi \rightarrow \psi$ is a decision rule and ϕ and ψ are P-basic and Q-basic formulas respectively, then the decision rule $\phi \rightarrow \psi$ is called a *PQ-basic decision rule* (in short *PQ*-rule).

A *PQ*-rule $\phi \rightarrow \psi$ is *admissible* in S if $\phi \wedge \psi$ is satisfiable in S.

Proposition 2.23 *A PQ-rule is true (consistent) in S iff all $\{P, Q\}$-basic formulas which occurr in the $\{P, Q\}$-normal form of the predecessor of the rule, also occurr in $\{P, Q\}$-normal form of the successor of the rule; otherwise the rule is false (inconsistent).*

The rule $b_0 \rightarrow c_2$ is false in the above table for *KR*-system 1, since the $\{b, c\}$-normal form of b_0 is $b_0c_2 \vee b_0c_3$, $\{b, c\}$-normal form of c_2 is b_0c_2, and the formula b_0c_3 does not occur in the successor of the rule.

On the other hand, the rule $a_2 \rightarrow c_3$ is true in the table, because the $\{a, c\}$-normal form of a_2 is a_2c_3, whereas the $\{a, c\}$-normal form of c_3 is $a_2c_3 \vee a_1c_3$.

Any finite set of decision rules in a DL-language is referred to as a *decision algorithm* in the DL-language. If all decision rules in a basic decision algorithm are *PQ*-decision rules, then the algorithm is said to be *PQ-decision algorithm*, or in short *PQ*-algorithm, and will be denoted by (P, Q).

A *PQ*-algorithm is *admissible* in S, if the algorithm is the set of all *PQ*-rules admissible in S.

A *PQ*-algorithm is *complete* in S, iff for every $x \in U$ there exists a *PQ*-decision rule $\phi \rightarrow \psi$ in the algorithm such that $x \models \phi \wedge \psi$ in S; otherwise the algorithm is *incomplete* in S.

A *PQ*-algorithm is *consistent* in S iff all its decision rules are consistent (true) in S; otherwise the algorithm is *inconsistent*.

Sometimes consistency (inconsistency) may be interpreted as *determinism* (*indeterminism*).

Given a *KR*-system, any two arbitrary, non-empty subset of attributes P, Q in the system determines uniquely a *PQ*-decision algorithm.

Consider the following *KR*-system from Pawlak [1].

Assume that $P = \{a, b, c\}$ and $Q = \{d, e\}$ are condition and decision attributes, respectively. Set P and Q uniquely associate the following *PQ*-decision algorithm with the table.

Table 2.2 *KR*-system 2

U	a	b	c	d	e
1	1	0	2	1	1
2	2	1	0	1	0
3	2	1	2	0	2
4	1	2	2	1	1
5	1	2	0	0	2

$$a_1b_0c_2 \rightarrow d_1e_1$$
$$a_2b_1c_0 \rightarrow d_1e_0$$
$$a_2b_1c_2 \rightarrow d_0e_2$$
$$a_1b_2c_2 \rightarrow d_1e_1$$
$$a_1b_2c_0 \rightarrow d_0e_2$$

If assume that $R = \{a, b\}$ and $T = \{c, d\}$ are condition and decision attributes, respectively, then the RT-algorithm determined by Table 2.2 is the following:

$$a_1b_0 \rightarrow c_2d_1$$
$$a_2b_1 \rightarrow c_0d_1$$
$$a_2b_1 \rightarrow c_2d_0$$
$$a_1b_2 \rightarrow c_2d_1$$
$$a_1b_2 \rightarrow c_0d_0$$

Of course, both algorithms are admissible and complete.

In order to check whether or not a decision algorithm is consistent, we have to check whether all its decision rules are true. The following proposition gives a much simpler method to solve this problem.

Proposition 2.24 *A PQ-decision rule $\phi \rightarrow \psi$ in a PQ-decision algorithm is consistent (true) in S iff for any PQ-decision rule $\phi' \rightarrow \psi'$ in PQ-decision algorithm, $\phi = \phi'$ implies $\psi = \psi'$.*

In Proposition 2.24, order of terms is important, since we require equality of expressions. Note also that in order to check whether or not a decision rule $\phi \rightarrow \psi$ is true we have to show that the predecessor of the rule (the formula ϕ) discerns the decision class ψ from the remaining decision classes of the decision algorithm in question. Thus, the concept of truth is somehow replaced by the concept of indiscernibility.

Consider the *KR*-system 2 again. With $P = \{a, b, c\}$ and $Q = \{d, e\}$ as condition and decision attributes. Let us check whether the *PQ*-algorithm:

Table 2.3 *KR*-system 2

U	a	b	c	d	e
1	1	0	2	1	1
4	1	2	2	1	1
2	2	1	0	1	0
3	2	1	2	0	2
5	1	2	0	0	2

$$a_1b_0c_2 \to d_1e_1$$
$$a_2b_1c_0 \to d_1e_0$$
$$a_2b_1c_2 \to d_0e_2$$
$$a_1b_2c_2 \to d_1e_1$$
$$a_1b_2c_0 \to d_0e_2$$

is consistent or not. Because the predecessors of all decision rules in the algorithm are different (i.e., all decision rules are discernible by predecessors of all decision rules in the algorithm), all decision rules in the algorithm are consistent (true) and cosequently the algorithm is consistent.

This can be also seen directly from Table 2.3.

The RT-algorithm, where $R = \{a, b\}$ and $T\{c, d\}$

$$a_1b_0 \to c_2d_1$$
$$a_2b_1 \to c_0d_1$$
$$a_2b_1 \to c_2d_0$$
$$a_1b_2 \to c_2d_1$$
$$a_1b_2 \to c_0d_0$$

is inconsistent bacause the rules

$$a_2b_1 \to c_0d_1$$
$$a_2b_1 \to c_2d_0$$

have the same predecessors and different successors, i.e., we are unable to discern c_0d_1 and c_2d_0 by means of condition a_2b_1. Thus, both rules are inconsistent (false) in the *KR*-system. Similarly, the rules

$$a_1b_2 \to c_2d_1$$
$$a_1b_2 \to c_0d_0$$

are also inconsistent (false).

We turn to *dependency* of attributes. Formally, the dependency is defined as below. Let $K = (U, \mathbf{R})$ be a knowledge base and $\mathbf{P}, \mathbf{Q} \subseteq \mathbf{R}$.

(1) Knowledge $\mathbf{Q}$ *depends on knowledge* $\mathbf{P}$ iff $IND(\mathbf{P}) \subseteq IND(\mathbf{Q})$.
(2) Knowlledge $\mathbf{P}$ and $\mathbf{Q}$ are *equivalent*, denoted $\mathbf{P} \equiv \mathbf{Q}$, if $\mathbf{P} \Rightarrow \mathbf{Q}$ and $\mathbf{Q} \Rightarrow \mathbf{P}$.
(3) Knowledge $\mathbf{P}$ and $\mathbf{Q}$ are *independent*, denoted $\mathbf{P} \not\equiv \mathbf{Q}$, iff neither $\mathbf{P} \Rightarrow \mathbf{Q}$ nor $\mathbf{Q} \Rightarrow \mathbf{P}$ hold.

Obviously, $\mathbf{P} \equiv \mathbf{Q}$ iff $IND(\mathbf{P}) \equiv IND(\mathbf{Q})$.

The dependency can be interpreted in different ways as Proposition 2.25 indicates:

Proposition 2.25 *The following conditions are equivalent:*

(1) $\mathbf{P} \Rightarrow \mathbf{Q}$
(2) $IND(\mathbf{P} \cup \mathbf{Q}) = INS(\mathbf{P})$
(3) $POS_{\mathbf{P}}(\mathbf{Q}) = U$
(4) $\underline{\mathbf{P}}X$ *for all* $X \in U/\mathbf{Q}$

where $\underline{\mathbf{P}}X$ *denotes* $\underline{IND(\mathbf{P})}/X$.

By Proposition 2.25, we can see the following: if $\mathbf{Q}$ depends on $\mathbf{P}$ then knowledge $\mathbf{Q}$ is superflous within the knowledge base in the sense that the knowledge $\mathbf{P} \cup \mathbf{Q}$ and $\mathbf{P}$ provide the same characterization of objects.

Proposition 2.26 *If* $\mathbf{P}$ *is a reduct of* $\mathbf{Q}$*, then* $\mathbf{P} \Rightarrow \mathbf{Q} - \mathbf{P}$ *and* $IND(\mathbf{P}) = IND(\mathbf{Q})$.

Proposition 2.27 *The following hold.*

(1) If $\mathbf{P}$ *is dependent, then there exists a subset* $\mathbf{Q} \subset \mathbf{P}$ *such that* $\mathbf{Q}$ *is a reduct of* $\mathbf{P}$.
(2) If $\mathbf{P} \subseteq \mathbf{Q}$ *and* $\mathbf{P}$ *is independent, then all basic relations in* $\mathbf{P}$ *are pairwise independent.*
(3) If $\mathbf{P} \subseteq \mathbf{Q}$ *and* $\mathbf{P}$ *is independent, then every subset* $\mathbf{R}$ *of* $\mathbf{P}$ *is independent.*

Proposition 2.28 *The following hold:*

(1) If $\mathbf{P} \Rightarrow \mathbf{Q}$ *and* $\mathbf{P}' \supset \mathbf{P}$*, then* $\mathbf{P}' \Rightarrow \mathbf{Q}$.
(2) If $\mathbf{P} \Rightarrow \mathbf{Q}$ *and* $\mathbf{Q}' \subset \mathbf{Q}$*, then* $\mathbf{P} \Rightarrow \mathbf{Q}'$.
(3) $\mathbf{P} \Rightarrow \mathbf{Q}$ *and* $\mathbf{Q} \Rightarrow \mathbf{R}$ *imply* $\mathbf{P} \Rightarrow \mathbf{R}$.
(4) $\mathbf{P} \Rightarrow \mathbf{R}$ *and* $\mathbf{Q} \Rightarrow \mathbf{R}$ *imply* $\mathbf{P} \cup \mathbf{Q} \Rightarrow \mathbf{R}$.
(5) $\mathbf{P} \Rightarrow \mathbf{R} \cup \mathbf{Q}$ *imply* $\mathbf{P} \Rightarrow \mathbf{R}$ *and* $\mathbf{P} \cup \mathbf{Q} \Rightarrow \mathbf{R}$.
(6) $\mathbf{P} \Rightarrow \mathbf{Q}$ *and* $\mathbf{Q} \cup \mathbf{R} \Rightarrow \mathbf{T}$ *imply* $\mathbf{P} \cup \mathbf{R} \Rightarrow \mathbf{T}$
(7) $\mathbf{P} \Rightarrow \mathbf{Q}$ *and* $\mathbf{R} \Rightarrow \mathbf{T}$ *imply* $\mathbf{P} \cup \mathbf{R} \Rightarrow \mathbf{Q} \cup \mathbf{T}$.

The derivation (dependency) can be partial, which means that only part of knowledge $\mathbf{Q}$ is derivable from knowledge $\mathbf{P}$. We can define the partial derivability using the notion of the positive region of knowledge.

Let $K = (U, \mathbf{R})$ be the knowledge base and $\mathbf{P}, \mathbf{Q} \subset \mathbf{R}$. Knowledge $\mathbf{Q}$ depends in a degree k $(0 \le k \le 1)$ from knowledge $\mathbf{P}$, in symbol $\mathbf{P} \Rightarrow_k \mathbf{Q}$, iff

$$k = \gamma_{\mathbf{P}}(\mathbf{Q}) = \frac{card(POS_{\mathbf{P}}(\mathbf{Q}))}{card(U)}$$

where $card$ denotes cardinality of the set.

If $k = 1$, we say that **Q** *totally depends from* **P**; if $0 < k < 1$, we say that **Q** *roughly (partially) depends from* **P**, and if $k = 1$ we say that **Q** is *totally independent from* **P**, If $\mathbf{P} \Rightarrow_1 \mathbf{Q}$, we shall also write $\mathbf{P} \Rightarrow \mathbf{Q}$.

The above ideas can also be interpreted as an ability to classify objects. More precisely, if $k = 1$, then all elements of the universe can be classified to elementary categories of $U/\mathbf{Q}$ by using knowledge **P**.

Thus, the coefficient $\gamma_{\mathbf{P}}(\mathbf{Q})$ can be understood as a degree of dependency between, **Q** and **P**. In other words, if we restrict the set of objects in the knowledge base to the set $POS_{\mathbf{P}}(\mathbf{Q})$, we would obtain the knowledge base in which $\mathbf{P} \Rightarrow \mathbf{Q}$ is a total dependency.

The measure k of dependency $\mathbf{P} \Rightarrow_k \mathbf{Q}$ does not capture how this partial dependency is actually distributed among classes of $U/\mathbf{Q}$. For example, some decision classes can be fully characterized by **P**, whereas others may be characteriaed only partially.

We will also need a coefficient $\gamma(X) = card(\underline{\mathbf{P}}X)/card(X)$ where $X \in U/\mathbf{Q}$ which shows how many elements of each class of $U/\mathbf{Q}$ can be classified by emplying knowledge **P**.

Thus, the two numbers $\gamma(\mathbf{Q})$ and $\gamma(X)$, $X \in U/\mathbf{Q}$ give us full information about "classification power" of the knowledge **P** with respect to the classification $U/\mathbf{Q}$.

Proposition 2.29 *The following hold:*

(1) If $\mathbf{R} \Rightarrow_k \mathbf{P}$ *and* $\mathbf{Q} \Rightarrow_l \mathbf{P}$*, then* $\mathbf{R} \cup \mathbf{Q} \Rightarrow_m \mathbf{P}$*, for some* $m \geq max(k, l)$.
(2) If $\mathbf{R} \cup \mathbf{P} \Rightarrow_k \mathbf{Q}$ *then* $\mathbf{R} \Rightarrow_l \mathbf{Q}$ *and* $\mathbf{P} \Rightarrow_m \mathbf{Q}$*, for some* $l, m, \leq k$.
(3) If $\mathbf{R} \Rightarrow_k \mathbf{Q}$ *and* $\mathbf{R} \Rightarrow_l \mathbf{P}$*, then* $\mathbf{R} \Rightarrow_m \mathbf{Q} \cup \mathbf{P}$*, for some* $m \leq max(k, l)$.
(4) If $\mathbf{R} \Rightarrow_k \mathbf{Q} \cup \mathbf{P}$*, then* $\mathbf{R} \Rightarrow_l \mathbf{Q}$ *and* $\mathbf{R} \Rightarrow_m \mathbf{P}$*, for some* $l, m \geq k$.
(5) If $\mathbf{R} \Rightarrow_k \mathbf{P}$ *and* $\mathbf{P} \Rightarrow_l \mathbf{Q}$*, then* $\mathbf{R} \Rightarrow_m \mathbf{Q}$*, for some* $m \geq l + k - 1$.

Here, we return to the decision algorithm for dependency. We say that the set of attributes Q depends *totally*, (or in short depends) on the set of attributes P in S, if there exists a consistent PQ-algorithm in S. If Q depends on P in S, we write $P \Rightarrow_S S$, or in short $P \Rightarrow Q$.

We can also define partial dependency of attributes. We say that the set of attributes Q *depends partially* on the set of attributes P in S, if there exists an inconsistent PQ-algorithm in S.

The degree of *dependency* between attributes can be defind. Let (P, Q) be a PQ-algorithm in S. By a *positive region* of the algorithm (P, Q), denoted $POS(P, Q)$, we mean the set of all consistent (true) PQ-rules in the algorithm.

The positive region of the decision algorithm (P, Q) is the consistent part (possibly empty) of the inconsistent algorithm. Obviously, a PQ-algorithm is inconsistent iff $POS(P, Q) \neq (P, Q)$ or what is the same $card(POS(P, Q)) \neq card(P, Q)$.

With every PQ-decision algorithm, we can associate a number
$k = card(POS(P, Q))/card(P, Q)$, called the degree of consistency, of the algorithm, or in short the degree of the algorithm, and we say that the PQ-algorithm has the degree (of consistency) k.

Obviously, $0 \leq k \leq 1$. If a *PQ*-algorithm has degree k, we can say that the set of attributes Q *depend in degree* k on the set of attributes P, denoted $P \Rightarrow_k Q$.

Naturally, the algorithm is consistent iff $k = 1$; otherwise, i.e., if $k \neq 1$, the algorithm. All these concepts are the same as in those discussed above. Note that in the consistent algorithm all decisions are uniquely determined by conditions in the decision algorithm. In other words, this means that all decisions in a consistent algorithm are discernible by means of conditions available in the decision algorithm.

Decision logic provides a simple means for reasoning about knowledge only by using propositional logic, and is suitable to some applications. Note here that the so-called *decision table* can serve as a *KR*-system.

However, the usuability of decision logic seems to be restrictive. In other words, it is far from a general system for reasoning in general. In this book, we will lay general frameworks for reasoning based on rough set theory.

2.6 Reduction of Knowledge

One of the important problems in rough set theory is whether the whole knowledge is always necessary to define some categories available in the knowledge considered. This problem is called *knowledge reduction*.

There are two basic concepts in reduction of knowledge, i.e., *reduct* and *core*. Intuitively, a reduct of knowledge is its essential part, which is sufficient to define all basic concepts in the considered knowledge. The core is the set of the most characteristic part of knowledge.

Let $\mathbf{R}$ be a family of equivalence relations and let $R \in \mathbf{R}$. We say that R is *dispensable* in $\mathbf{R}$ if $IND(\mathbf{R}) = IND(\mathbf{R} - \{R\})$; otherwise is *indispensable* in $\mathbf{R}$. The family $\mathbf{R}$ is *independent* if each $R \in \mathbf{R}$ is indispensable in $\mathbf{R}$; otherwise $\mathbf{R}$ is *dependent*.

Proposition 2.30 *If* $\mathbf{R}$ *is independent and* $\mathbf{P} \subseteq \mathbf{R}$*, then* $\mathbf{P}$ *is also independent.*

The following proposition states the relationship between the core and reducts.

$\mathbf{Q} \subseteq \mathbf{P}$ is a *reduct* of $\mathbf{P}$ if $\mathbf{Q}$ is indenpendent and $IND(\mathbf{Q}) = IND(\mathbf{P})$. Obviously, $\mathbf{P}$ may have many reducts. The set of all indispensable relations in $\mathbf{P}$ is called the *core* of $\mathbf{P}$ denoted $CORE(\mathbf{P})$.

Proposition 2.31 $CORE(\mathbf{P}) = \bigcap RED(\mathbf{P})$*, where* $RED(\mathbf{P})$ *is the famiy of all reducts of* $\mathbf{P}$.

Here is an example from Pawlak [1]. Suppose $\mathbf{R} = \{P, Q, R\}$ of three equivalence relations P, Q and R with the following equivalence classes:

$$U/P = \{\{x_1, x_4, x_5\}, \{x_2, x_8\}, \{x_3\}, \{x_6, x_7\}\}$$
$$U/Q = \{\{x_1, x_3, x_5\}, \{x_6\}, \{x_2, x_4, x_7, x_8\}\}$$
$$U/R = \{\{x_1, x_5\}, \{x_6\}, \{x_2, x_7, x_8\}, \{x_3, x_4\}\}$$

Thus, the relation $IND(\mathbf{R})$ has the equivalence classes:

$$U/IND(\mathbf{R}) = \{\{x_1, x_5\}, \{x_2, x_8\}, \{x_3\}, \{x_4\}, \{x_6\}, \{x_7\}\}.$$

The relation P is indispensable in $\mathbf{R}$, since

$$U/IND(\mathbf{R} - \{P\}) = \{\{x_1, x_3\}, \{x_2, x_7, x_8\}, \{x_3\}, \{x_4\}, \{x_6\}\} \neq U/IND(\mathbf{R})$$

For relation Q, we have:

$$U/IND(\mathbf{R} - \{Q\}) = \{\{x_1, x_3\}, \{x_2, x_8\}, \{x_3\}, \{x_4\}, \{x_6\}, \{x_7\}\} = U/IND(\mathbf{R})$$

thus the relation Q is dispensable in $\mathbf{R}$.

Similarly, for relation R, we have:

$$U/IND(\mathbf{R} - \{R\}) = \{\{x_1, x_3\}, \{x_2, x_8\}, \{x_3\}, \{x_4\}, \{x_6\}, \{x_7\}\} = U/IND(\mathbf{R})$$

hence the relation R is also dispensable in $\mathbf{R}$.

Thus, the classification defined by the set of three equivalence relations P, Q and R is the same as the classification defined by relation P and Q or P and R.

To find reducts of the family $\mathbf{R} = \{P, Q, R\}$, we have to check whether pairs of relations P, Q and P, R are independent or not. Because $U/IND(\{P, Q\}) \neq U/IND(P)$ and $U/IND(\{P, Q\}) \neq U/IND(Q)$, the relations P and Q are independent. Consequently, $\{P, Q\}$ is a reduct of $\mathbf{R}$. Similarly, we can see that $\{P, R\}$ is also a reduct of $\mathbf{R}$.

Thus, there are two reducts of the family $\mathbf{R}$, namely $\{P, Q\}$ and $\{P, R\}$, and $\{P, Q\} \cap \{P, R\} = \{P\}$ is the core of $\mathbf{R}$.

The concepts of reduct and core defined above can be generalized. Let P and Q be equivalence relations over U. *P-positive region* of Q, denoted $POS_P(Q)$, is defined as follows:

$$POS_P(Q) = \bigcup_{X \in U/Q} \underline{P}X$$

The positive region of Q is the set of all objects of the universe U which can be properly classified to classes of U/Q employing knowledge expressed by the classification U/P.

Let $\mathbf{P}$ and $\mathbf{Q}$ be families of equivalence relations over U. We say that $R \in \mathbf{P}$ is $\mathbf{Q}$-dispensable in $\mathbf{P}$, if

$$POS_{IND(\mathbf{P})}(IND(\mathbf{Q})) = POS_{IND(\mathbf{P}-\{R\})}(IND(\mathbf{Q}))$$

otherwise R is $\mathbf{Q}$-indispensable in $\mathbf{P}$.

If every R in $\mathbf{P}$ is $\mathbf{Q}$-indispensable, we say that $\mathbf{P}$ is $\mathbf{Q}$-independent. The family $\mathbf{S} \subseteq \mathbf{P}$ is called a $\mathbf{Q}$-reduct of $\mathbf{P}$ iff $\mathbf{S}$ is the $\mathbf{Q}$-independent subfamily of $\mathbf{P}$ and

$POS_{\mathbf{S}}(\mathbf{Q}) = POS_{\mathbf{P}}(\mathbf{Q})$. The set of all **Q**-indispensable elementary relation in **P** is called the **Q**-core, denoted $CORE_{\mathbf{Q}}(\mathbf{P})$.

The following proposition shows the relationship of relative reduct and core.

Proposition 2.32 $CORE_{\mathbf{Q}}(\mathbf{P}) = \bigcap RED_{\mathbf{Q}}(\mathbf{P})$, *where* $RED_{\mathbf{Q}}$ *is the family of all* **Q**-*reducts of* **P**.

Let $POS_{\mathbf{P}}(\mathbf{Q})$ is the set of all objects which can be classified to elementary categories of knowledge **Q** employing knowledge **Q**. Knowledge **P** is **Q**-independent if the whole knowledge **P** is necessary to classify objects to elementary categories of knowledge **Q**.

The **Q**-core knowledge of knowldge **P** is the essential part of knowledge **P**, which cannot be eliminated without disturbing the ability to classify objects to elementary categories of **Q**.

The **Q**-reduct of knowledge **P** is the minimal subset of knowledge **P**, which provides the same classification of objects to elementary categories of knowledge **Q** as the whole knowledge **P**. Note that knowledge **P** can have more than one reduct.

Knowledge **P** with only one **Q**-reduct is, in a sense, deterministic, i.e., there is only one way of using elementary categories of knowledge **P** when classifying objects to elementary categories of knowledge **Q**.

If knowledge **P** has many **Q**-reducts, then it is non-deterministic, and there are in general many ways of using elementary categories of **P** when clasifying objects to elementary categories of **Q**.

This non-determinism is particularly strong if the core knowledge is void. But non-determinism introduces synonymy to the knowledge, which in some cases may be a drawback.

We turn to reduction of categories. Basic categories are pieces of knowledge, which can be considered as "building blocks" of concepts. Every concept in the knowledge base can be only expessed (exactly or approximately) in terms of basic categories.

On the other hand, every basic category is "build up" (is an intersection) of some elementary categories. Then, the question arises whether all the elementary categories are necessary to define the basic categories in question.

The problem can be formulated precisely as follows. Let $F = \{X_1, ..., X_n\}$ be a family of sets such that $X_i \subseteq U$. We say that X_i is *dispensable*, if $\bigcap(F - \{X_i\}) = \bigcap F$; otherwise the set X_i is *indispensable* in F.

The family F is *independent* if all its complements are indispensable in F; otherwise F is *dependent*. The family $H \subseteq F$ is a *reduct* of F, if H is independent and $\bigcap H = \bigcap F$. The family of all indispensable sets in F is called the *core* of F, denoted $CORE(F)$.

Proposition 2.33 $CORE(F) = \bigcap RED(F)$, *where* $RED(F)$ *is the family of all reducts of* F.

Now, we introduce the example from Pawlak [1], Let the family of three sets be $F = \{X, Y, Z\}$, where

$$X = \{x_1, x_3, x_8\}$$
$$Y = \{x_1, x_3, x_4, x_5, x_6\}$$
$$Z = \{x_1, x_3, x_4, x_6, x_7\}.$$

Hence, $\bigcap F = X \cap Y \cap Z = \{x_1, x_3\}$. Bacause

$$\bigcap(F - \{X\}) = Y \cap Z = \{x_1, x_3, x_4, x_6\}$$
$$\bigcap(F - \{Y\}) = X \cap Z = \{x_1, x_3\}$$
$$\bigcap(F - \{Z\}) = X \cap Y = \{x_1, x_3\}$$

sets Y and Z are dispensable in the family F and the family F is dependent. Set X iw the core of F. Families $\{X, Y\}$ and $\{X, Z\}$ are reducts of F and $\{X, Y\} \cap \{X, Z\} = \{X\}$ is the core of F.

We also need a method to elminate superfluous categories from categories which are the union of some categories. The problem can be formulated in a way similar to the previous one, with the exception that now instead of intersection of sets we will need union of sets.

Let $F = \{X_1, ..., X_n\}$ be a family of sets such that $X_i \subseteq U$. We say that X_i is *dispensable* in $\bigcup F$, if $\bigcup(F - |X_i\}) = \bigcup F$; otherwise the set X_i is *indispensable* in $\bigcup F$.

The family F is *independent* with respect to $\bigcup F$ if all its components are indispensable in $\bigcup F$; otherwise F is *dependent* in $\bigcup F$. The family $H \subseteq F$ is a *reduct* of $\bigcup F$, if H is independent with respect to $\bigcup H$ and $\bigcup H = \bigcup F$.

Here is th example from Pawlak [1]. Let $F = \{X, Y, Z, T\}$, where

$$X = \{x_1, x_3, x_8\}$$
$$Y = \{x_1, x_2, x_4, x_5, x_6\}$$
$$Z = \{x_1, x_3, x_4, x_6, x_7\}$$
$$T = \{x_1, x_2, x_5, x_7\}$$

Obviously, $\bigcup F = X \cup Y \cup Z \cup T = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$.

Because we have:

$$\bigcup(F - \{X\}) = \bigcup\{Y, Z, T\} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\} \neq \bigcup F$$
$$\bigcup(F - \{Y\}) = \bigcup\{X, Z, T\} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\} = \bigcup F$$
$$\bigcup(F - \{Z\}) = \bigcup\{X, Y, T\} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\} = \bigcup F$$
$$\bigcup(F - \{T\}) = \bigcup\{X, Y, Z\} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\} = \bigcup F$$

thus the only indispensable set in the family F is the set X, and remaining sets Y, Z and T are dispensable in the family.

Hence, the following sets are reducts of F: $\{X, Y, Z\}$, $\{X, Y, T\}$, $\{X, Z, T\}$. That means that the concept $\bigcup F = X \cup Y \cup Z \cup T$, which is the union of X, Y, Z and T can be simplified and represented as union of smaller numbers of concepts.

Here, we discuss the relative reduct and the core of categories. Suppose that $F = \{X_1, ..., X_n\}$, $X_i \subseteq U$ and a subset $Y \subseteq U$ such that $\bigcap F \subseteq Y$.

We say that X_i is *Y-dispensable* in $\bigcap F$, if $\bigcap (F - \{X_i\}) \subseteq Y$; otherwise the set X_i is *Y-indispensable* in $\bigcap F$.

The family F is *Y-independent* in $\bigcap F$, if all its complements are Y-indispensable in $\bigcap F$; otherwise F is *Y-dependent* in $\bigcup F$.

The family $H \subseteq F$ is a *Y-reduct* of $\bigcap F$, if H is Y-independenct in $\bigcap F$ otherwise F and $\bigcap H \subseteq Y$.

The family of all Y-indispensable sets in $\bigcap F$ is called the *Y-core* of F, denoted $CORE_F(F)$. We also say that a Y-reduct (Y-core) is a relative reduct (core) with respect to Y.

Proposition 2.34 *$CORE_Y(F) = \bigcap RED_Y(F)$, where $RED_Y(F)$ is the family of all Y-reducts of F.*

Thus, superfluous elementary categories can be eliminated from the basic categories in a similar way as the equivalence relations.

As discussed above, reduction of knowledge is to remove superfluous partitions (equivalence relations). For this task, the concept of reduct and core play significant roles.

2.7 Knowledge Representation

In this section, we discuss a *knowledge representation system* (KR system), which can be seen as a formal language. It is interpreted as data table and plays an important role in practical applications.

A knowledge representation system is a pair $S = (U, A)$, where U is a non-empty finite set called *universe* and A is a non-empty finite set of *primitive attributes*. Every primitive attribute $a \in A$ is a total function $a : U \rightarrow V_a$, where V_a is the set of values of a, called the *domain* of a.

For every subset of attributes $B \subseteq A$, we associate a binary relation $IND(B)$, called an *indiscernibility relation*, defined as:

$$IND(B) = \{(x, y) \in U^2 \mid for\ every\ a \in B, a(x) = a(y)\}$$

Obviously, $IND(B)$ is an equivalence relation and the following holds.

$$IND(B) = \bigcap_{a \in B} IND(a)$$

Every subset $B \subseteq A$ is called an *attribute*. If B is a single element set, then B is called *primitive*, otherwise *compound*.

Attribute B may be considered as a name of the relation $IND(B)$, or in other words, a name of knowledge represented by an equivalence relation $IND(B)$.

Thus, the knowledge representation system $S = (U, A)$ may be viewed as a description of a knowledge base $K = (U, \mathbf{R})$. Here, each equivalence relation in the knowledge base is represented by an attribute and each equivalence class of the relation by an attribute value.

It is noted that there is a one-to-one correspondence between knowledge bases and knowledge representation systems. To check it, it suffices to assign to arbitrary knowledge base $K = (U, \mathbf{R})$ a knowledge representation system $S = (U, A)$ in the following way.

If $R \in \mathbf{R}$ and $U/R = \{X_1, ..., X_k\}$, then to the set of attributes A every attribute $a_R : U \to V_{a_R}$ such that $V_{a_R} = \{1, ..., k\}$ and $a_R(x) = i$ iff $x \in X$ for $i = 1, .., k$. Then, all notions of knowledge bases can be expressed in terms of notions of knowledge representation systems.

Consider the following knowledge representation system from Pawlak [1]:

Here, the universe U consists of 8 elements numbered 1, 2, 3, 4, 5, 6, 7 and 8, the set of attributes is $A = \{a, b, c, d, e\}$, whereas $V = V_a = V_b = V_c = V_d = V_e = \{0, 1, 2\}$.

In Table 2.4, elements 1, 4 and 5 of U are indiscernible by attribute a, elements 2, 7 and 8 are indiscernible by the set of attriutes $\{b, c\}$, and elements 2 and 7 are indiscernible by the set of attributes $\{d, e\}$.

Partitions generated by attributes in this system are given below:

$$U/IND\{a\} = \{\{2, 8\}, \{1, 4, 5\}, \{3, 6, 7\}\}$$
$$U/IND\{b\} = \{\{1, 3, 5\}, \{2, 4, 7, 8\}, \{6\}\}$$
$$U/IND\{c, d\} = \{\{1\}, \{3, 6\}, \{2, 7\}, \{4\}, \{5\}, \{8\}\}$$
$$U/IND\{a, b, c\} = \{\{1, 5\}, \{2, 8\}, \{3\}, \{4\}, \{6\}, \{7\}\}$$

Table 2.4 KR-system 3

U	a	b	c	d	e
1	1	0	2	2	0
2	0	1	1	1	2
3	2	0	0	1	1
4	1	1	0	2	2
5	1	0	2	0	1
6	2	2	0	1	1
7	2	1	1	1	2
8	0	1	1	0	1

For example, for the set of attributes $C = \{a, b, c\}$ and the subset $X = \{1, 2, 3, 4, 5\}$ of the universe, we have $\underline{C}X = \{1, 2, 3, 4, 5\}$, $\overline{C}X = \{1, 2, 3, 4, 5, 8\}$ and $BN_C(X) = \{2, 8\}$.

Thus, the set X is rough with respect to the attribute C, which is to say that we are unable to decide whether elements 2 and 8 are members of X or not, employing the set of attributes C. For the rest of the universe classification of elements using the set C of attributes is possible.

The set of attributes $C = \{a, b, c\}$ is dependent. The attributes a and b are indispensable, whereas the attribute c is superfluous. Here, the dependency $\{a, b\} \Rightarrow \{c\}$ holds. Because $IND\{a, b\}$ has the blocks $\{1, 5\}, \{2, 8\}, \{3\}, \{4\}, \{6\}, \{7\}$, and $IND\{c\}$ has the blocks $\{1, 5\}, \{2, 7, 8\}, \{3, 4, 6\}$, $IND\{a, b\} \subset IND\{c\}$.

We next compute the degree of dependency of attribute $D = \{d, e\}$ from the attributes $C = \{a, b, c\}$ in Table 2.4. The partition $U/IND(C)$ consists of the blocks, $X_1 = \{1\}, X_2 = \{2, 7\}, X_3 = \{3, 6\}, X_4 = \{4\}, X_5 = \{5, 8\}$. The partition $U/IND(D)$ consists of the blocks, $Y_1 = \{1, 5\}, Y_2 = \{2, 8\}, Y_3 = \{3\}, Y_4 = \{4\}, Y_5 = \{6\}, Y_6 = \{7\}$.

Because $\underline{C}X_1 = \emptyset, \underline{C}X_2 = Y_6, \underline{C}X_3 = Y_3 \cup Y_5, \underline{C}X_4 = Y_4$ and $\underline{C}X_5 = \emptyset$, we have $POS(D) = Y_3 \cup Y_4 \cup Y_5 \cup Y_6 = \{3, 4, 6, 7\}$.

Namely, only these elements can be classified into blocks of the partition $U/IND(D)$ employing the set $C = \{a, b, c\}$ attributes. Hence, the degree of dependency between C and D is $\gamma_C(D) = 4/8 = 0.5$.

The set of attributes C is D-independent, and the attribute a is D-indispensable. This means that the the D-core of C is one attribute set $\{a\}$. Thus, there are the following dependencies: $\{a, b\} \Rightarrow \{d, e\}$ and $\{a, c\} \Rightarrow \{d, e\}$ in the table.

When speaking about attributes, it is obvious that they may have varying importance in the analysis of considered issues. To find out the significance of a specific attribute (or group of attributes) it seems reasonable to drop the attribute from the table and see how classification will be changed without this attribute.

If removing the attribute will change the classification considerably it means that its significance is high-in the opposite case, the significance should be low. The idea can be precisely employing the concept of a positive region.

As a measure of the signignificance of the subset of attributes $B' \subseteq B$ with respect to the classification induced by a set of attributes C, we will mean the difference:

$$\gamma_B(C) - \gamma_{B-B'}(C)$$

which expresses how the positive region of the classification $U/IND(C)$ when classifying the object by means of attribute B will be affect if we drop some attributes (subset B') from the set B.

Let us compute the significance of the attributes a, b and c with respect to the set of attributes $\{d, e\}$ in Table 2.4. $POS_C(D) = \{3, 4, 6, 7\}$, where $C = \{a, b, c\}$ and $D = \{d, e\}$. Because

$$U/IND\{b,c\} = \{\{1,5\},\{2,7,8\},\{3\},\{4\},\{6\}\}$$
$$U/IND\{a,c\} = \{\{1,5\},\{2,8\},\{3,6\},\{4\},\{7\}\}$$
$$U/IND\{a,b\} = \{\{1,5\},\{2,8\},\{3\},\{4\},\{6\},\{7\}\}$$
$$U/IND\{d,e\} = \{\{1\},\{2,7\},\{3,6\},\{4\},\{5,8\}\}$$

we have:

$$POS_{C-\{a\}}(D) = \{3,4,6\}$$
$$POS_{C-\{b\}}(D) = \{3,4,6,7\}$$
$$POS_{C-\{c\}}(D) = \{3,4,6,7\}$$

Consequently, corresponding accuracies are:

$$\gamma_{C-\{a\}}(D) = 0.125,$$
$$\gamma_{C-\{b\}}(D) = 0,$$
$$\gamma_{C-\{c\}}(D) = 0.$$

Thus, the attribute a is most significant, since it most changes the positive region of $U/IND(D)$, i.e., without the attribute a we are unable to classify object 7 to classes of $U/IND(D)$.

Note that the attribute a is D-indispensable and the attributes b and c are D-dispensable. Thus, the attribute a is the core of C with respect to D (D-core of C) and $\{a,b\}$ and $\{a,c\}$ are reducts of C with respect to D (D-reducts of C).

Knowledge representation systems can be expressed by means of tables, but as will be discussed in Chap. 4 it can be also formalized in the framework of modal logic.

We may find some similarities between knowledge representation systems and *relational databases* (cf. Codd [15]) in that the concept of table plays a crucial role. There is, however, an essential difference between these two models.

Most importantly, the relational model is not interested in the meaning of the information stored in the table. It focusses on efficient data structuring and manipulation. Consequently, the objects about which information is contained in the table are not represented in the table.

On the other hand, in knowledge representation system, all objects are explicitly represented and the attribute values, i.e., the table entries, have associated explicit meaning as features or properties of the objects.

2.8 Decision Tables

Decision tables can be seen as a special, important class of knowledge representation systems, and can be used for applications. Let $K = (U, A)$ be a knowledge representation system and $C, D \subset A$ be two subsets of attributes, called *condition* and *decision* attributes, respectively.

KR-system with distinguished condition and decision attributes is called a *decision table*, denoted $T = (U, A, C, D)$ or in short DC. Equivalence clases of the relations $IND(C)$ and $IND(D)$ are called *condition* and *decision classes*, respectively.

With every $x \in U$, we associate a function $d_x : A \to V$, such that $d_x(a) = a(x)$ for every $a \in C \cup D$; the function d_x is called a *decision rule* (in T), and x is referred as a *label* of the decision rule d_x.

Note that elements of the set U in a decision table do not represent in general any real objects, but are simple identifiers of decision rules.

If d_x is a decision rule, then the restriction of d_x to C, denoted $d_x \mid C$, and the restriction of d_x to D, denoted $d_x \mid D$ are called *conditions* and *decisions* (actions) of d_x, respectively.

The decision rule d_x is *consistent* (in T) if for every $y \neq x, d_x \mid C = d_y \mid C$ implies $d_x \mid D = d_y \mid D$; otherwise the decision rule is *inconsistent*.

A decision table is *consistent* if all its decision rules are consistent; otherwise the decision table is *inconsistent*. Consistency (inconsistency) sometimes may be interpreted as determinism (non-determinism).

Proposition 2.35 *A decision table $T = (U, A, C, D)$ is consistent iff $C \Rightarrow D$.*

From Proposition 2.35, it follows that the practical method of checking consistency of a decision table is by simply computing the degree of dependency between condition and decision attributes. If the degree of dependency equals to 1, then we conclude that the table is consistent; otherwise it is inconsistent.

Proposition 2.36 *Each decision table $T = (U, A, C, D)$ can be uniquely decomposed into two decision tables $T_1 = (U, A, C, D)$ and $T_2 = (U, A, C, D)$ such that $C \Rightarrow_1 D$ in T_1 and $C \Rightarrow_0 D$ in T_2 such that $U_1 = POS_C(D)$ and $U_2 = \bigcup_{X \in U/IND(D)} BN_C(X)$.*

Proposition 2.36 states that we can decompose the table into two subtables; one totally inconsistent with dependency coefficient equal to 0, and the second entirely consistent with the dependency equal to 1. This decomposition however is possible only if the degree of dependency is greater than 0 and different from 1.

Consider Table 2.5 from Pawlak [1].

Assume that a, b and c are condition attributes, and d and e are decision attributes. In this table, for instance, the decision rule 1 is inconsistent, whereas the decision rule 3 is consistent. By Proposition 2.36, we can decompose Decision Table 1 into the following two tables:

Table 2.2 is consistent, whereas Table 2.3 is totally inconsistent, which mens all decision rules in Table 2.2 are consistent, and in Table 2.3 all decision rules are inconsistent.

Simplification of decision tables is very important in many applications, e.g. software engineering. An example of simplification is the redeuction of condition attributes in a decision table.

Table 2.5 Decision Table 1

U	a	b	c	d	e
1	1	0	2	2	0
2	0	1	1	1	2
3	2	0	0	1	1
4	1	1	0	2	2
5	1	0	2	0	1
6	2	2	0	1	1
7	2	1	1	1	2
8	0	1	1	0	1

In the reduced decision table, the same decisions can be based on a smaller number of conditions. This kind of simplification eliminates the need for checking unnecessary conditions.

Pawlak proposed simplification of decision tables which includes the following steps:

(1) Computation of reducts of condition attributes which is equivalent to elimination of some column from the decision table.
(2) Elimination of duplicate rows.
(3) Elimination of superfluous values of attributes.

Thus, the method above consists in removing superfluous condition attributes (columns), duplicate rows and, in addition to that, irrelevant values of condition attributes.

By the above procedure, we obtain an "incomplete" decision table, containing only those values of condition attributes which are necessary to make decisions. According to our definition of a decision table, the incomplete table is not a decision table and can be treated as an abbreviation of such a table.

For the sake of simplicity, we assume that the set of condition attribute is already reduced, i.e., there are not superfluous condition attributes in the decision table.

With every subset of attributes $B \subseteq A$, we can associate partition $U/IND(B)$ and consequently the set of condition and decision attributes define partitions of objects into condition and decision classes.

We know that with every subset of attributes $B \subseteq A$ and object x we may associate set $[x]_B$, which denotes an equivalence class of the relation $IND(B)$ containing an object x, i.e., $[x]_B$ is an abbreviation of $[x]_{IND(B)}$.

Thus, with any set of condition attributes C in a decision rule d_x we can associate set $[x]_C = \cap_{a \in C}[x]_a$. But, each set $[x]_a$ is uniquely determined by attribute value $a(x)$. Hence, in order to remove superfluous values of condition attributes, we have to eliminate all superfluous equivalence classes $[x]_a$ from the equivalence class $[x]_C$. Thus, problems of elimination of superfluous values of attributes and elimination of corresponding equivalence classes are equivalent.

Consider the following decision table from Pawlak [1].

Here, a, b and c are condition attributes and e is a decision attribute.

It is easy to compute that the only e-dispensable condition attribute is c; consequently, we can remove column c in Table 2.4, which yields Table 2.5:

In the next step, we have to reduce superfluous values of condition attributes in every decision rule. First, we have to compute core values of condition attributes in every decision rule.

Here, we compute the core values of condition attributes for the first decision rule, i.e., the core of the family of sets

$$\mathbf{F} = \{[1]_a, [1]_b, [1]_d\} = \{\{1, 2, 4, 5\}, \{1, 2, 3\}, \{1, 4\}\}$$

From this we have:

$$[1]_{\{a,b,d\}} = [1]_a \cap [1]_b \cap [1]_d = \{1, 2, 4, 5\} \cap \{1, 2, 3\} \cap \{1, 4\} = \{1\}.$$

Moreover, $a(1) = 1, b(1) = 0$ and $d(1) = 1$. In order to find dispensable categories, we have to drop one category at a time and check whether the intersection of remaining categories is still included in the decision category $[1]_e = \{1, 2\}$, i.e.,

$$[1]_b \cap [1]_d = \{1, 2, 3\} \cap \{1, 4\} = \{1\}$$
$$[1]_a \cap [1]_d = \{1, 2, 4, 5\} \cap \{1, 4\} = \{1, 4\}$$
$$[1]_a \cap [1]_b = \{1, 2, 4, 5\} \cap \{1, 2, 3\} = \{1, 2\}$$

This means that the core value is $b(1) = 0$. Similarly, we can compute remaining core values of condition attributes in every decision rule and the final results are represented in Table 2.6.

Then, we can proceed to compute value reducts. As an example, let us compute value reducts for the first decision rule of the decision table.

Accordingly to the definition of it, in order to compute reducts of the family $\mathbf{F} = \{[1]_a, [1]_b, [1]_d\} = \{\{1, 2, 3, 5\}, \{1, 2, 3\}, \{1, 4\}\}$, we have to find all subfamilies $\mathbf{G} \subseteq \mathbf{F}$ such that $\bigcap \mathbf{G} \subseteq [1]_e = \{1, 2\}$. There are four following subfamilies of $\mathbf{F}$:

$$[1]_b \cap [1]_d = \{1, 2, 3\} \cap \{1, 4\} = \{1\}$$
$$[1]_a \cap [1]_d = \{1, 2, 4, 5\} \cap \{1, 4\} = \{1, 4\}$$
$$[1]_a \cap [1]_b = \{1, 2, 4, 5\} \cap \{1, 2, 3\} = \{1\}$$

Table 2.6 Decision Table 2.2

U	a	b	c	d	e
3	2	0	0	1	1
4	1	1	0	2	2
6	2	2	0	1	1
7	2	1	1	1	2

and only two of them

$$[1]_b \cap [1]_d = \{1, 2, 3\} \cap \{1, 4\} = \{1\} \subseteq [1]_e = \{1, 2\}$$
$$[1]_a \cap [1]_b = \{1, 2, 4, 5\} \cap \{1, 2, 3\} = \{1\} \subseteq [1]_e = \{1, 2\}$$

are reducts of the family **F**. Hence, we have two values reducts: $b(1) = 0$ and $d(1) = 1$ or $a(1) = 1$ and $b(1) = 0$. This means that the attribute values of attributes a and b or d and e are characteristic for decision class 1 and do not occur in any other decision classes in the decision table. We see also that the value of attribute b is the intersection of both value reducts, $b(1) = 0$, i.e., it is the core value.

In Table 2.7, we list value reducts fir all decision rules in Table 2.1.

Seen from Decision Table 2.7, for decision rules 1 and 2 we have two value reducts of condition attributes. Decision rules 3,4 and 5 have only one value reducts of condition attributes for each decision rule row. The remaining decision rules 6 and 7 contain two and three value reducts, respectively.

Hence, there are two reduced form of decision rule 1 and 2, decision rule 3, 4 and 5 have only one reduced form each, decision rule 6 has two reducts and decision rule 7 has three reducts.

Thus, there are $4 \times 2 \times 3 = 24$ (not necessarily different) solutions to our problem. One such solution is presented in Decision Table 2.8.

Another solution is shown in Decision Table 2.9.

Because decision rules 1 and 2 are identical, and so are rules 5, 6 and 7, we can represent Decision Table 2.10:

In fact, enumeration of decision rules is not essential, so we can enumerate them arbitrary and we get as a final result Decision Table 2.11:

Table 2.7 Decision Table 2.3

U	a	b	c	d	e
1	1	0	2	2	0
2	0	1	1	1	2
5	1	0	2	0	1
8	0	1	1	0	1

Table 2.8 Decision Table 2.4

U	a	b	c	d	e
1	1	0	0	1	1
2	1	0	0	0	1
3	0	0	0	0	0
4	1	1	0	1	0
5	1	1	0	2	2
6	2	1	0	2	2
7	2	2	2	2	2

Table 2.9 Decision Table 2.5

U	a	b	d	e
1	1	0	1	1
2	1	0	0	1
3	0	0	0	0
4	1	1	1	0
5	1	1	2	2
6	2	1	2	2
7	2	2	2	2

Table 2.10 Decision Table 2.6

U	a	b	d	e
1	_	0	_	1
2	1	_	_	1
3	0	_	_	0
4	_	1	1	0
5	_	_	2	2
6	_	_	_	2
7	_	_	_	2

Table 2.11 Decision Table 2.7

U	a	b	d	e
1	1	0	$\times$	1
$1'$	$\times$	0	1	1
2	1	0	$\times$	1
$2'$	1	$\times$	0	1
3	0	$\times$	$\times$	0
4	$\times$	1	1	0
5	$\times$	$\times$	2	2
6	$\times$	$\times$	2	2
$6'$	2	$\times$	$\times$	2
7	$\times$	$\times$	2	2
$7'$	$\times$	2	$\times$	2
$7''$	2	$\times$	$\times$	2

This solution is referred to as *minimal*. The presented method of decision table simplification can be named *semantic*, since it refers to the meaning of the information contained in the table. Another decision table simplification is also possible and can be named *syntactic*. It is described within the framework of decision logic (Table 2.12).

To simplify a decision table, we should first find reducts of condition attributes, remove duplicate rows and then find value-reducts of condition attributes and again, if necessary, remove duplicate rows (Table 2.13).

This method leads to a simple algorithm for decision table simplification, Note that a subset of attributes may have more than one reduct (relative reduct). Thus, the simplification of decision table does not yield unique results. Some decision tables possibly can be optimized according to preassumed criteria (Table 2.14).

Table 2.12 Decision Table 2.8

U	*a*	*b*	*d*	*e*
1	1	0	×	1
2	1	×	0	1
3	0	×	×	0
4	×	1	1	0
5	×	×	2	2
6	×	×	2	2
7	2	×	×	2

Table 2.13 Decision Table 2.9

U	*a*	*b*	*d*	*e*
1	1	0	×	1
2	1	0	×	1
3	0	×	×	0
4	×	1	1	0
5	×	×	2	2
6	×	×	2	2
7	×	×	2	2

Table 2.14 Decision Table 2.10

U	*a*	*b*	*d*	*e*
1, 2	1	0	×	1
3	0	×	×	0
4	×	1	1	0
5, 6, 7	×	×	2	2

Table 2.15 Decision Table 2.11

U	a	b	d	e
1	1	0	×	1
2	0	×	×	0
3	×	1	1	0
4	×	×	2	2

We have finished the presentation of some topics in rough set theory. Pawlak also established other formal results about rough sets and discusses advantages of rough set theory. We here omit these issues; see Pawlak [1] (Table 2.15).

References

1. Pawlak, P.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
2. Ziarko, W.: Variable precision rough set model. J. Comput. Syst. Sci. **46**, 39–59 (1993)
3. Pawlak, P.: Rough sets. Int. J. Comput. Inf. Sci. **11**, 341–356 (1982)
4. Shen, Y., Wang, F.: Variable precision rough set model over two universes and its properties. Soft. Comput. **15**, 557–567 (2011)
5. Zadeh, L.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
6. Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets Syst. **1**, 3–28 (1976)
7. Negoita, C., Ralescu, D.: Applications of Fuzzy Sets to Systems Analysis. Wiley, New York (1975)
8. Nakamura, A., Gao, J.: A logic for fuzzy data analysis. Fuzzy Sets Syst. **39**, 127–132 (1991)
9. Quafafou, M.: α-RST: a generalizations of rough set theory. Inf. Sci. **124**, 301–316
10. Cornelis, C., De Cock, J., Kerre, E.: Intuitionistic fuzzy rough sets: at the crossroads of imperfect knowledge. Expert Syst. **20**, 260–270 (2003)
11. Atnassov, K.: Intuitionistic Fuzzy Sets. Physica, Haidelberg (1999)
12. Ganter, B., Wille, R.: Formal Concept Analysis. Springer, Berlin (1999)
13. Ore, O.: Galois connexion. Trans. Am. Math. Soc. **33**, 493–513 (1944)
14. Armstrong, W.: Dependency structures in data base relationships, IFIP'74, pp. 580–583 (1974)
15. Codd, E.: A relational model of data for large shared data banks. Commun. ACM **13**, 377–387 (1970)

Springer

http://www.springer.com/978-3-319-72690-8

Reasoning with Rough Sets
Logical Approaches to Granularity-Based Framework
Akama, S.; Murai, T.; Kudo, Y.
2018, X, 201 p. 12 illus., 7 illus. in color., Hardcover
ISBN: 978-3-319-72690-8