Rough Sets: Theoretical aspects of reasoning about data

wu

March 11, 2021

Contents

1 Knowledge

1.1 Knowledge base

Given a finite set $U \neq \emptyset$ (the universe). Any subset $X \subset U$ of the universe is called a **concept** or a **category** in U. And any family of concepts in U will be referred to as **abstract knowledge** about U.

partition or **classification** of a certain universe U is a family $C = \{X_1, X_2, \dots, X_n\}$ s.t. $X_i \subset U, X_i \neq \emptyset, X_i \cap X_j = \emptyset$ and $\bigcup X_i = U$

A family of classifications is called a **knowledge base** over U

R an equivalence relation over U, U/R family of all equivalence classes of R, referred to be **categories** or **concepts** of R, and $[x]_R$ denotes a category in R containing an element $x \in U$

By a **knowledge base** we can understand a relational system $K = (U, \mathbf{R})$, $\mathbf{R}$ is a family of equivalence relations over U

If $\mathbf{P} \subset \mathbf{R}$ and $\mathbf{P} \neq \emptyset$, then $\bigcap \mathbf{P}$ is also an equivalence relation, and will be denoted by $IND(\mathbf{P})$, called an **indiscernibility relation** over $\mathbf{P}$

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

$U/IND(\mathbf{P})$ called **P-basic knowledge about** U in K. For simplicity, $U/\mathbf{P} = U/IND(\mathbf{P})$ and **P** will be also called **P-basic knowledge** . Equivalence classes of $IND(\mathbf{P})$ are called **basic categories** of knowledge **P**. If $Q \in \mathbf{R}$, then Q is a Q**-elementary knowledge** and equivalence classes of Q are referred to as Q**-elementary concepts** of knowledge **R**

The family of all **P**-basic categories for all $\neq \mathbf{P} \subset \mathbf{R}$ will be called the **family of basic categories** in knowledge base $K = (U, \mathbf{R})$

Let $K = (U, \mathbf{R})$ be a knowledge base. By $IND(K)$ we denote the family of all equivalence relations defined in K as $IND(K) = \{IND(\mathbf{P}) : \emptyset \neq \mathbf{P} \subseteq \mathbf{R}\}$.

Thus $IND(K)$ is the minimal set of equivalence relations.

Every union of **P**-basic categories will be **P-category**

The family of all categories in the knowledge base $K = (U, \mathbf{R})$ will be referred to as K**-categories**

1.2 Equivalence, generalization and specialization of knowledge

Let $K = (U, \mathbf{P}), K' = (U, \mathbf{Q})$. K and K' are **equivalent** $K \simeq K'$, $(\mathbf{P} \simeq \mathbf{Q})$ if $IND(\mathbf{P}) = IND(\mathbf{Q})$. Hence $K \simeq K'$ if both K and K' have the same set of elememtary categories. *This means that knowledge in knowledge bases K and K' enables us to express exactly the same facts about the universe.*

If $IND(\mathbf{P}) \subset IND(\mathbf{Q})$ then knowledge **P** is **finer** than knowledge **Q** (**coarser**). **P** is **specialization** of **Q** and **Q** is **generalization** of **P**

2 Imprecise categories, approximations and rough sets

2.1 Rough sets

Let $X \subseteq U$. X is R**-definable** or R**-exact** if X is the union of some R-basic categories. otherwise R**-undefinable**, R**-rough**, R**-inexact** .

2.2 Approximations of set

Given $K = (U, \mathbf{R}), R \in IND(K)$

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

called the R**-lower** and R**-upper approximation** of X

$BN_R(X) = \overline{R}X - \underline{R}X$ is R-**boundary** of X. $BN_R(X)$ is the set of elements which cannot be classified either to X or to $-X$ having knowledge R

$$
\begin{aligned}
&POS_R(X) = \underline{R}X, R\text{-positive region of } X \\
&NEG_R(X) = U - \overline{R}X, R\text{-negative region of } X \\
&BN_R(X) - R\text{-borderline region of } X
\end{aligned}
$$

If $x \in POS(X)$, then x will be called an R-**positive example of** X

Proposition 2.1. *1. X is R-definable if and only if $\underline{R}X = \overline{R}X$*

2. X is rought w.r.t. R if and only if $\underline{R}X \neq \overline{R}X$

2.3 Properties of approximations

Proposition 2.2 (2.2). *1. $\underline{R}X \subseteq X \subseteq \overline{R}X$*

2. $\underline{R}\emptyset = \underline{R}\emptyset = \emptyset$; $\underline{R}U = \overline{R}U = U$

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. $\underline{R}(X \cup Y) \subseteq \underline{R}X \cup \underline{R}Y$

8. $\underline{R}(-X) = -\overline{R}X$

9. $\overline{R}(-X) = -\underline{R}X$

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 equivalence relation R over U uniquely defines a topological space $T = (U, DIS(R))$ where $DIS(R)$ is the familty of all open and closed set in T and U/R is a base for T. The R-lower and R-upper approximation of X in A are **interior** and **closure** operations in the topological space T

2.4 Approximations and membership relation

$$x \underline{\in}_R X \text{if and only if } x \in \underline{R}X$$
$$x \overline{\in}_R X \text{if and only if } x \in \overline{R}X$$

where $\underline{\in}_R$ read "x **surely belongs** to X w.r.t. R" and $\overline{\in}_R$ - "x **possibly belongs** to X w.r.t. R". The **lower** and **upper** membership.

Proposition 2.3.

1. *$x \underline{\in} X$ implies $x \in X$ implies $x \overline{\in} X$*
2. *$X \subset Y$ implies ($x \underline{\in} X$ implies $x \underline{\in} Y$ and $x \overline{\in} X$ implies $x \overline{\in} Y$)*
3. *$x \overline{\in} (X \cup Y)$ if and only if $x \overline{\in} X$ or $x \overline{\in} Y$*
4. *$x \underline{\in} (X \cap Y)$ if and only if $x \underline{\in} X$ and $x \underline{\in} Y$*
5. *$x \underline{\in} X$ or $x \underline{\in} Y$ implies $x \underline{\in} (X \cup Y)$*
6. *$x \overline{\in} X \cap Y$ implies $x \overline{\in} X$ and $x \overline{\in} Y$*
7. *$x \underline{\in} (-X)$ if and only if non $x \overline{\in} X$*
8. *$x \overline{\in} (-X)$ if and only if non $x \underline{\in} X$*

2.5 Numerical characterization of imprecision

accuracy measure

$$\alpha_R(X) = \frac{card\ \underline{\underline{R}}}{card\ \overline{\overline{R}}}$$

2.6 Topological characterization of imprecision

Definition 2.4.

1. If $\underline{R}X \neq \emptyset$ and $\overline{R}X \neq U$, then we say that X is **roughly R-definable**. We can decide whether some elements belong to X or $-X$
2. If $\underline{R}X = \emptyset$ and $\overline{R}X \neq U$, then we say that X is **internally R-undefinable**. We can decide whether some elemnts belong to $-X$
3. If $\underline{R}X \neq \emptyset$ and $\overline{R}X = U$, then we say that X is **externally R-undefinable**. We can decide whether some elements belong to X

4. If $\underline{R}X = \emptyset$ and $\overline{R}X = U$, then we say that X is **totally R-undefinable**. unable to decide

Proposition 2.5 (2.4). *1. Set X is R-definable(roughly R-definable, totally R-undefinable) if and only if so is $-X$*

2. Set X is externally R-undefinable if and only if $-X$ is internally R-undefinable

Proof. 1.

$$\begin{aligned} R\text{-definable} &\Leftrightarrow \underline{R}X = \overline{R}X, \underline{R} \neq \emptyset, \overline{R} \neq U \\ &\Leftrightarrow -\underline{R}X = -\overline{R}X \\ &\Leftrightarrow \overline{R}(-X) = \underline{R}(-X) \end{aligned}$$

$$\begin{aligned} X \text{ is roughly } R\text{-definable} &\Leftrightarrow \underline{R}X \neq \emptyset \wedge \overline{R}X \neq U \\ &\Leftrightarrow -\underline{R}X \neq U \wedge -\overline{R}X \neq \emptyset \\ &\Leftrightarrow \overline{R}(-X) \neq U \wedge \underline{R}(-X) \neq \emptyset \end{aligned}$$

□

2.7 Approximation of classifications

If $F = \{X_1, \dots, X_n\}$ is a family of non empty sets, then $\underline{R}F = \{\underline{R}X_1, \dots, \underline{R}X_n\}$ and $\overline{R}F = \{\overline{R}X_1, \dots, \overline{R}X_n\}$, called the R-**lower approximation** and the R-**upper approximation** of the family F

The **accuracy of approximation** of F by R is

$$\alpha_R(F) = \frac{\sum card\ \underline{R}X_i}{\sum card\ \overline{R}X_i}$$

quality of approximation of F by R

$$\gamma_R(F) = \frac{\sum card\ \underline{R}X_i}{card\ U}$$

Proposition 2.6 (2.5). *Let $F = \{X_1, \dots, X_n\}$ where $n > 1$ be a classification of U and let R be an equivalence relation. If there exists $i \in \{1, 2, \dots, n\}$ s.t. $\underline{R}X_i \neq \emptyset$, then for each $j \neq i$ and $j \in \{1, \dots, n\}$, $\overline{R}X_j \neq U$*

Proof. If $\underline{R}X_i \neq \emptyset$ then there exists $x \in X$ s.t. $[x]_R \subseteq X$, which implies $[x]_R \cap X_j = \emptyset$ for each $j \neq i$. This yields $\overline{R}X_j \cap [x]_R = \emptyset$. □

Proposition 2.7 (2.6). *Let $F = \{X_1, \dots, X_n\}, n > 1$ be a classification of U and let R be an equivalence relation. If there exists $i \in \{1, \dots, n\}$ s.t. $\overline{R}X_i = U$, then for each $j \neq i$ and $j \in \{1, \dots, n\}$ $\underline{R}X_j = \emptyset$*

Proposition 2.8 (2.7). *Let $F = \{X_1, \dots, X_n\}, n > 1$ be a classification of U and let R be an equivalence relation. If for each $i \in \{1, 2, \dots, n\}$ $\underline{R}X_i \neq \emptyset$ holds, then $\overline{R}X_i \neq U$ for each $i \in \{1, \dots, n, \}$*

Proposition 2.9. *Let $F = \{X_1, \dots, X_n\}, n > 1$ be a classification of U and let R be an equivalence relation. If for each $i \in \{1, 2, \dots, n\}$ $\overline{R}X_i = U$ holds, then $\underline{R}X_i = \emptyset$ for each $i \in \{1, \dots, n\}$*

2.8 Rough equality of sets

Definition 2.10. Let $K = (U, \mathbf{R})$ be a knowledge base, $X, Y \subseteq U$ and $R \in IND(K)$, then

1. Sets X and Y are **bottom** R**-equal** $(X \eqsim_R Y)$ 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 $X \simeq_R Y$ and $X \eqsim_R Y$

Proposition 2.11 (2.9). *1. $X \eqsim Y$ iff $X \cap Y \eqsim X$ and $X \cap Y \eqsim Y$*

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 $X \eqsim X'$ and $Y \eqsim Y'$ then $X \cap Y \eqsim X' \cap Y'$*
5. *If $X \simeq Y$, then $X \cup -Y \simeq U$*
6. *If $X \eqsim Y$, then $X \cap -Y \eqsim \emptyset$*
7. *If $X \subseteq Y$ and $Y \simeq \emptyset$, then $X \simeq \emptyset$*
8. *If $X \subseteq Y$ and $X \subseteq U$ then $Y \subseteq U$*
9. *$X \simeq Y$ iff $-X \eqsim -Y$*
10. *If $X \eqsim \emptyset$ or $Y \eqsim \emptyset$, then $X \cap Y \eqsim \emptyset$*

11. *If $X \simeq U$ or $Y \simeq U$, then $X \cup Y \simeq U$*

Proposition 2.12 (2.10)**.** *For any equivalence relation R*

1. *$\underline{R}X$ is the intersection of all $Y \subseteq U$ s.t. $X \eqsim_R Y$*
2. *$\overline{R}$ is the union of all $Y \subseteq U$ s.t. $X \simeq_R Y$*

2.9 Rough inclusion of sets

Definition 2.13. Let $K = (U, \mathbf{R})$ be a knowledge base, $X, Y \subseteq U$ and $R \in IND(K)$.

1. Set X is **bottom R-included** in Y ($X \underset{\sim}{\subset}_R Y$) 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 ($X \underset{\sim}{\overset{\sim}{\subset}}_R Y$) iff $X \overset{\sim}{\subset}_R Y$ and $X \underset{\sim}{\subset}_R Y$

Proposition 2.14 (2.11)**.** 1. *If $X \subseteq Y$, then $X \underset{\sim}{\subset} Y, X \overset{\sim}{\subset} Y$ and $X \underset{\sim}{\overset{\sim}{\subset}} Y$*

2. *If $X \underset{\sim}{\subset} Y$ and $Y \underset{\sim}{\subset} X$, then $X \eqsim Y$*
3. *If $X \overset{\sim}{\subset} Y$ and $Y \overset{\sim}{\subset} X$, then $X \simeq Y$*
4. *If $X \underset{\sim}{\overset{\sim}{\subset}} Y$ and $Y \underset{\sim}{\overset{\sim}{\subset}} X$ then $X \approx Y$*
5. *$X \overset{\sim}{\subset} Y$ iff $X \cup Y \simeq Y$*
6. *$X \underset{\sim}{\subset} Y$ iff $X \cap Y \eqsim X$*
7. *If $X \subseteq Y, X \eqsim X', Y \eqsim Y'$, then $X' \underset{\sim}{\subset} Y'$*
8. *If $X \subseteq Y, X \simeq X', Y \simeq Y'$, then $X' \overset{\sim}{\subset} Y'$*
9. *If $X \subseteq Y, X \approx X', Y \approx Y'$, then $X' \underset{\sim}{\overset{\sim}{\subset}} Y'$*
10. *If $X' \overset{\sim}{\subset} X$ and $Y' \overset{\sim}{\subset} Y$, then $X' \cup Y' \overset{\sim}{\subset} X \cup Y$*
11. *If $X' \underset{\sim}{\subset} X, Y' \underset{\sim}{\subset}$ then $X' \cap Y' \underset{\sim}{\subset} X \cap Y$*
12. *$X \cap Y \underset{\sim}{\subset} X \overset{\sim}{\subset} X \cup Y$*
13. *If $X \underset{\sim}{\subset} Y$ and $X \eqsim Z$ then $Z \underset{\sim}{\subset} Y$*
14. *If $X \overset{\sim}{\subset} Y$ and $X \simeq Z$ then $Z \overset{\sim}{\subset} Y$*
15. *If $X \underset{\sim}{\overset{\sim}{\subset}} Y$ and $X \approx$ then $Z \underset{\sim}{\overset{\sim}{\subset}} Y$*

3 Reduction of knowledge

3.1 Reduct and Core of Knowledge

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

Proposition 3.1 (3.1)**.** *If $\mathbf{R}$ is independent and $\mathbf{P} \subseteq \mathbf{R}$, then $\mathbf{P}$ is also independent*

Proof. $IND(\mathbf{R}) = IND(\mathbf{P} \cup (\mathbf{R} - \mathbf{P})) = IND(\mathbf{P}) \cap IND(\mathbf{R} - \mathbf{P})$ □

$\mathbf{Q} \subseteq \mathbf{R}$ is a **reduct** of $\mathbf{P}$ if $\mathbf{Q}$ is independent and $IND(\mathbf{Q}) = IND(\mathbf{P})$

The set of all indispensable relations in $\mathbf{P}$ is called the **core** of $\mathbf{P}$ denoted by $CORE(\mathbf{P})$

Proposition 3.2 (3.2)**.**

$$CORE(\boldsymbol{P}) = \bigcap RED(\boldsymbol{P})$$

where $RED(\boldsymbol{P})$ is the family of all reducts of $\boldsymbol{P}$

Proof. If $\mathbf{Q}$ is a reduct of $\mathbf{P}$ and $R \in \mathbf{P} - \mathbf{Q}$, then $IND(\mathbf{P}) = IND(\mathbf{Q})$. If $\mathbf{Q} \subseteq \mathbf{R} \subseteq \mathbf{P}$ then $IND(\mathbf{Q}) = IND(\mathbf{R})$. Assuming $\mathbf{R} = \mathbf{P} - \{R\}$ then $R \notin CORE(\mathbf{P})$

If $R \notin CORE(\mathbf{P})$. This means $IND(\mathbf{P}) = IND(\mathbf{P} - \{R\})$ which implies that there exists an independent subset $\mathbf{S} \subseteq \mathbf{P} - \{R\}$ s.t. $IND(\mathbf{S}) = IND(\mathbf{P})$. Hence $R \notin \bigcap RED(\mathbf{P})$ □

3.2 Relative reduct and relative core of knowledge

Let P and Q be equivalence relations over U

P**-positive**

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

The P-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

$R \in \mathbf{P}$ is **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 will say that $\mathbf{P}$ is **$\mathbf{Q}$ -independent** or $\mathbf{P}$ is **independent w.r.t. $\mathbf{Q}$**

The family $\mathbf{S} \subseteq \mathbf{P}$ will be called a **$\mathbf{Q}$ -reduct** of $\mathbf{P}$ if and only if $\mathbf{S}$ is the $\mathbf{Q}$ -independent subfamily of $\mathbf{P}$ and $POS_{\mathbf{S}}(\mathbf{Q}) = POS_{\mathbf{P}}(\mathbf{Q})$

The set of all $\mathbf{Q}$ -indispensable elmentary relations in $\mathbf{P}$ will be called the **$\mathbf{Q}$ -core** of $\mathbf{P}$ and will be denoted as $CORE_{\mathbf{Q}}(\mathbf{P})$

Proposition 3.3 (3.3)**.**

$$CORE_{\mathbf{Q}}(\boldsymbol{P}) = \bigcap RED_{\mathbf{Q}}(\boldsymbol{P})$$

where $RED_{\mathbf{Q}}(\boldsymbol{P})$ is the family of all $\boldsymbol{Q}$ -reducts of $\boldsymbol{P}$

3.3 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 expressed (exactly or approximately) in terms of basic categories. On the other hand, every basic category is "built up" (is an intersection) of some elementary categories. Thus the question arises whether all the elementary categories are necessary to define the basic categories in question.

Let $F = \{X_1, \dots, X_n\}$ be a family of sets s.t. $X_i \subseteq U$.

X_i is **dispensable** in F if $\bigcap(F - \{X_i\}) = \bigcap F$, otherwise the set X_i is **indispensable** in F

The family F is **independent** if all of its components 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 will be called the **core** of F, denoted $CORE(F)$

Proposition 3.4 (3.4)**.**

$$CORE(F) = \bigcap RED(F)$$

3.4 Relative reduct and core of categories

$F = \{X_1, \dots, X_n\}$, $X_i \subseteq U$ and a subset $Y \subseteq U$ s.t. $\bigcap F \subseteq Y$

X_i is **Y-dispensable** in $\bigcap F$ if $\bigcap(F - \{X_i\}) \subseteq Y$. Otherwise X_i is **Y-indispensable**

The family F is **Y-independent** in $\bigcap F$ if all of its components are **Y-indispensable** in $\bigcap F$

The family $H \subseteq F$ is a Y**-reduct** of $\bigcap F$ if H is Y-independent in $\bigcap F$ and $\bigcap H \subseteq Y$

The family of all Y-indispensable sets in $\bigcap F$ will be called the Y **core** of F and will be denoted by $CORE_Y(F)$

Proposition 3.5 (3.5)**.**

$$CORE_Y(F) = \bigcap RED_Y(F)$$

4 Dependencies in knowledge base

4.1 Dependency of knowledge

Knowledge **Q** is **derivable** from knowledge **P** if all elementary categories of **Q** can be defined in terms of some elementary categories of knowledge **P**. If **Q** is derivable from **P** we will also say that **Q depends** on **P** and can be written $\mathbf{P} \Rightarrow \mathbf{Q}$

Let $K = (U, \mathbf{R})$ be a knowledge base and let $\mathbf{P}, \mathbf{Q} \subseteq \mathbf{R}$

1. Knowledge **Q depends on knowledge P** iff $IND(\mathbf{P}) \subseteq IND(\mathbf{Q})$ note that $IND(\mathbf{P})$ is a set of pair

2. Knowledge **P** and **Q** are **equivalent** denoted as $\mathbf{P} \equiv \mathbf{Q}$ iff $\mathbf{P} \Rightarrow \mathbf{Q}$ and $\mathbf{Q} \Rightarrow \mathbf{P}$

3. Knowledge **P** and **Q** are **independent** denoted as $\mathbf{P} \not\equiv \mathbf{Q}$ iff neither $\mathbf{P} \Rightarrow \mathbf{Q}$ nor $\mathbf{Q} \Rightarrow \mathbf{P}$

Obiviously $\mathbf{P} \equiv \mathbf{Q}$ if and only if $IND(\mathbf{P}) = IND(\mathbf{Q})$

Proposition 4.1 (4.1)**.** *The following conditions are equivalent*

1. $\boldsymbol{P} \Rightarrow \boldsymbol{Q}$

2. $IND(\boldsymbol{P} \cup \boldsymbol{Q}) = IND(\boldsymbol{P})$

3. $POS_{\boldsymbol{P}}(\boldsymbol{Q}) = POS_{IND(\boldsymbol{P})}(\boldsymbol{Q}) = U$

4. $\underline{\boldsymbol{P}}X = X$ *for all* $X \in U/Q$

where $\underline{\boldsymbol{P}}X$ *denotes* $\underline{IND(\boldsymbol{P})}X$

Proposition 4.2 (4.2)**.** *If* $\boldsymbol{P}$ *is a reduct of* $\boldsymbol{Q}$ *then* $\boldsymbol{P} \Rightarrow \boldsymbol{Q} - \boldsymbol{P}$ *and* $IND(\boldsymbol{P}) = IND(\boldsymbol{Q})$

Proof. 1. $(1) \to (2)$

$IND(\mathbf{P}) \subseteq IND(\mathbf{P} \cup \mathbf{Q}) \subseteq IND(\mathbf{P})$

2. $(2) \to (3)$

$$\begin{aligned} POS_{IND(\mathbf{P})}(\mathbf{Q}) &= \bigcup_{X \in U/\mathbf{Q}} \underline{IND(\mathbf{P})}X \\ &= \bigcup_{X \in U/\mathbf{Q}} \underline{IND(\mathbf{P} \cup \mathbf{Q})}X \end{aligned}$$

Since $\mathbf{Q} \subseteq \mathbf{P} \cup \mathbf{Q}$, $IND(\mathbf{P} \cup \mathbf{Q}) \subseteq IND(\mathbf{Q})$ and for each $x \in U$, $[x]_{IND(\mathbf{P} \cup \mathbf{Q})} \subseteq [x]_{IND(\mathbf{Q})}$, which means for any $Y \in U/\mathbf{P} \cup \mathbf{Q}$, there exists some $X \in U/\mathbf{Q}$ s.t. $Y \subseteq X$. Hence $POS_{\mathbf{P}}(\mathbf{Q}) = U$

3. $(3) \to (4)$

$$\begin{aligned} POS_{\mathbf{P}}(\mathbf{Q}) &= \bigcup_{X \in U/\mathbf{Q}} \underline{IND(\mathbf{P})}X \\ &= \bigcup_{X \in U/bQ} \underline{\mathbf{P}}X = U \end{aligned}$$

And $\underline{\mathbf{P}}X \subseteq X$

4. $(4) \to (1)$

$$\begin{aligned} \mathbf{P} \Rightarrow \mathbf{Q} &\Leftrightarrow IND(\mathbf{P}) \subseteq IND(\mathbf{Q}) \\ &\Leftrightarrow \forall x \in U, [x]_{IND(\mathbf{P})} \subseteq [x]_{IND(\mathbf{Q})} \end{aligned}$$

□

Proof. $\mathbf{P} \Rightarrow \mathbf{Q} - \mathbf{P} \Leftrightarrow IND(\mathbf{P} \cup \mathbf{Q} - \mathbf{P}) = IND(\mathbf{P})$ □

Proposition 4.3 (4.3). *1. If $\mathbf{P}$ is dependent, then there exists a subset $\mathbf{Q} \subset \mathbf{P}$ s.t. $\mathbf{Q}$ is a reduct of $\mathbf{P}$*

2. If $\mathbf{P} \subseteq \mathbf{Q}$ and $\mathbf{P}$ is dependent, 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 4.4 (4.4). *1. If $\mathbf{P} \Rightarrow \mathbf{Q}$ and $\mathbf{P}' \supset \mathbf{P}$, then $\mathbf{P}' \Rightarrow \mathbf{Q}$*

2. *If $\boldsymbol{P} \Rightarrow \boldsymbol{Q}$ and $\boldsymbol{Q}' \subset \boldsymbol{Q}$ then $\boldsymbol{P} \Rightarrow \boldsymbol{Q}'$*
3. *$\boldsymbol{P} \Rightarrow \boldsymbol{Q}$ and $\boldsymbol{Q} \Rightarrow \boldsymbol{R}$ imply $\boldsymbol{P} \Rightarrow \boldsymbol{R}$*
4. *$\boldsymbol{P} \Rightarrow \boldsymbol{R}$ and $\boldsymbol{Q} \Rightarrow \boldsymbol{R}$ imply $\boldsymbol{P} \cup \boldsymbol{Q} \Rightarrow \boldsymbol{R}$*
5. *$\boldsymbol{P} \Rightarrow \boldsymbol{R} \cup \boldsymbol{Q}$ implies $\boldsymbol{P} \Rightarrow \boldsymbol{R}$ and $\boldsymbol{P} \Rightarrow \boldsymbol{Q}$*
6. *$\boldsymbol{P} \Rightarrow \boldsymbol{Q}$ and $\boldsymbol{R} \Rightarrow \boldsymbol{T}$ imply $\boldsymbol{P} \cup \boldsymbol{R} \Rightarrow \boldsymbol{Q} \cup \boldsymbol{T}$*
7. *$\boldsymbol{P} \Rightarrow \boldsymbol{Q}$ and $\boldsymbol{R} \Rightarrow \boldsymbol{T}$ imply $\boldsymbol{P} \cup \boldsymbol{R} \Rightarrow \boldsymbol{Q} \cup \boldsymbol{T}$*

4.2 Partial dependency 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** $k (0 \leq k \leq 1)$ from knowledge $\mathbf{P}$, symbolically $\mathbf{P} \Rightarrow_k \mathbf{Q}$ if and only if

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

If $k = 1$, $\mathbf{Q}$ **totally depends from** $\mathbf{P}$. If $0 < k < 1$, $\mathbf{Q}$ **roughly depends from** $\mathbf{P}$. If $k = 0$, $\mathbf{Q}$ is **totally independent from** $\mathbf{P}$

Ability to classify objects.

Proposition 4.5 (4.5)**.**

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

5 Knowledege prepresentation

5.1 Formal definition

Knowledge representation system is a pair $S = (U, A)$ where U is a nonempty finite set called the **universe**, and A is a nonempty finite set of **primitive attributes**

Every primitive attribute $a \in A$ is a total function $a : U \to V_a$ where V_a is the **domain** of a

With every subset of attributes $B \subseteq A$ we associate a binary relation $IND(B)$ called and **indiscernibility relation**

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

$IND(B)$ is an euivalence relation and

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

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

$a(x)$ can be viewed as a name of $[x]_{IND(a)}$. The name of an elementary category of attribute $B \subseteq A$ containing object x is a set of pairs $\{a, a(x) : a \in B\}$

There is a one-to-one correspondence between knowledge bases and knowledge representation system up to isomorphism of attributes and attribute names

Suppose

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

The universe $U = \{1, 2, 3, 4, 5, 6, 7, 8\}$. $V = V_a = \cdots = V_e = \{0, 1, 2\}$

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

5.2 Discernibility matrix

Let $S = (U, A)$ be a knowledge representation system with $U = \{x_1, x_2, \dots, x_n\}$. By an **discernibility matrix of** S is

$$M(S) = (c_{ij}) = \{a \in A : a(x_i) \neq a(x_j)\} \quad \text{for } i, j = 1, 2, \dots, n$$

Now the core can be defined as the set of all single element entries of the discernibility matrix

$B \subseteq A$ is the reduct of A if B is the minimal subset of A s.t.

$$B \cap c \neq \emptyset \text{ for any nonempty entry } c (c \neq \emptyset) \text{ in } M(S)$$

6 Decision tables

6.1 Formal definition and some properties

Let $K = (U, A)$ be a knowledge representation system and let $C, D \subset A$ be two subsets of attributes called **condition** and **decision attributes** repectively. KR-system with distinguished condition ad decision attributes will be called a **decision table** and will be denoted by $T = (U, A, C, D)$ or in short CD

Equivalence classes of the relations $IND(C)$ and $IND(D)$ will be called **condition** and **decision classes**

With every $x \in U$ we associate a function $d_x : A \to V$ s.t. $d_x(a) = a(x)$ for every $a \in C \cup D$. The function d_x will be called a **decision rule**

If d_x is a decision rule, then the restriction of d_x to C, denoted $d_x|C$ and the restriction of d_x to D, denoted $d_x|D$ will be called **conditions** and **decisions** of d_x

The decision rule d_x is **consistent** if for every $y \neq x$, $d_x|C = d_y|C$ implies $d_x|D = d_y|D$. Otherwise **inconsistent**

A decision table is **consistent** if al its decision rules are consistent

Proposition 6.1 (6.1). *A decision table $T = (U, A, C, D)$ is consistent if and only if $C \Rightarrow D$*

Proposition 6.2 (6.2). *Each decision table $T = (U, A, C, D)$ can be uniquely decomposed into two decision tables $T_1 = (U, A, C, D)$ and $T_2 = (U, A, C, D)$ s.t. $C \Rightarrow_1 D$ in T_1 and $C \Rightarrow_0 D$ in T_2 where $U_1 = POS_C(D)$ and $U_2 = \bigcup_{X \in U/IND(D)} BN_C(X)$*

Example. Consider

Table 1: Knowledge representation system

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

Assume that a,b,c are condition attributes and d,e are decision attributes.

$$U/\{a\} = \{\{2,8\},\{1,4,5\},\{3,6,7\}\}$$
$$U/\{b\} = \{\{1,3,5\},\{2,4,7,8\},\{6\}\}$$
$$U/\{c\} = \{\{3,4,6\},\{2,7,8\},\{1,5\}\}$$
$$U/\{d\} = \{\{5,8\},\{2,3,6,7\},\{1,4\}\}$$
$$U/\{e\} = \{\{1\},\{3,5,6,8\},\{2,4,7\}\}$$
$$U/\{a,b,c\} = \{\{1,5\},\{2,8\},\{3\},\{4\},\{6\},\{7\}\}$$
$$U/\{d,e\} = \{\{1\},\{2,7\},\{3,6\},\{4\},\{5,8\}\}$$
$$POS_C(D) = \{3,4,6,7\}$$
$$\bigcup_{X \in /IND(D)} BN_C(X) = \{1,2,5,8\}$$

Table 2

U_1	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

Table 2 is consistent whereas table 3 is totally inconsistent

6.2 Simplification of decision tables

Step

Table 3

U_2	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

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

Suppose $B \subseteq A$ and an object x. $\forall C, [x]_C = \bigcup_{a \in C} [x]_a$. Each $[x]_a$ 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 $[x]_C$

Given

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

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

e-dispensable condition attribute is c and we can remove it

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

Next we need to reduce superfluous values of condition attributes. First compute core values of condition attributes

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

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

is

$$[1]_{\{a,b,d\}} = [1]_a \cap [1]_b \cap [1]_d = \{1\}$$

Moreover $a(1) = 1, b(1) = 0, d(1) = 1$. 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 $[1]_e = \{1,2\}$

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

a is dispensable. This means that the core value is $b(1) = 0$

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

Having computed core values of condition attributes, we can proceed to compute value reducts.

Only $[1]_b \cap [1]_d$ and $[1]_a \cap [1]_b$ are reducts of the family $\mathbf{F}$. Hence

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

Note that

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

we have

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

This solution is **minimal**

7 Reasoning about knowledge

7.1 The language of decision logic

alphabet of the language

1. A - the set of **attribute constant**

2. $V = \bigcup V_\alpha$, the set of **attribute value constants** $\alpha \in A$

3. Set $\{\sim, \wedge, \vee, \to, \equiv\}$ of propositional connectives, called **negation** …

Set of formulas

1. Expressions of the form (a, v) or in short a_v called **elementary formulas** are formulas of the DL-language for any $a \in A, v \in V_a$

2. If ϕ and ψ are formulas of the DL-language, then so are $\sim \phi$,$(\phi \vee \psi)$,$(\phi \wedge \psi)$,$(\phi \to \psi)$ and $(\phi \equiv \phi)$

7.2 Semantics of decision logic language

atomic formula (a, v) is interpreted as a description of all objects having value v for attribute a. By the model we mean the KR-system $S = (U, A)$. Thus the model S describes the meaning of symbols of predicates (a, v) in U

An object $x \in U$ **satisfies** a formula ϕ in $S = (U, A)$ denoted $x \vDash_S \phi$ or in short $x \vDash \phi$ if and only if

1. $x \vDash (a, v)$ iff $a(x) = v$

2. $x \vDash \sim \phi$ iff $x \nvDash \phi$

3. $x \vDash \phi \vee \psi$ iff $x \vDash \phi$ or $x \vDash \psi$

4. $x \vDash \phi \vee \psi$ iff $x \vDash \phi$ and $x \vDash \psi$

As a corollary from the above conditions we get

1. $x \vDash \phi \to \psi$ iff $x \vDash \sim \phi \vee \psi$

2. $x \vDash \phi \equiv \psi$ iff $x \vDash \phi \to \psi$ and $x \vDash \psi \to \phi$

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

$$|\phi|_S = \{x \in U : x \vDash_S \phi\}$$

called the **meaning** of the formula ϕ in S

Proposition 7.1 (7.1). *1.* $|(a, v)|_S = \{x \in U : a(x) = v\}$

2. $|\sim \phi|_S = -|\phi|_S$

3. $|\phi \vee \psi|_S = |\phi|_S \cup |\psi|_S$

4. $|\phi \wedge \psi|_S = |\phi|_S \cup |\psi|_S$

5. $|\phi \to \psi|_S = -|\phi|_S \cup |\psi|_S$

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

A formula ϕ is said to be **true** in a KR-system S, $\models_S \phi$ if and only if $|\phi|_S = U$

Formulas ϕ and ψ are equivalent in S if and only if $|\phi|_S = |\psi|_S$

Proposition 7.2 (7.2). *1.* $\models_S \phi$ *iff* $|\phi|_S = U$

2. $\models_S \sim \phi$ *iff* $|\phi|_S = \emptyset$

3. $\models_S \phi \to \psi$ *iff* $|\phi|_S \subseteq |\psi|_S$

4. $\models_S \phi \equiv \psi$ *iff* $|\phi|_S = |\psi|_S$

7.3 Deduction in decision logic

In order to define our logic, we need to verify the semantic equivalence of formulas. To do this we need to finish with suitable rules for transforming formulas without changing their meanings.

Abbreviations:

$$\phi \wedge \sim \phi =_{df} 0 \text{ and } \phi \vee \sim \phi =_{df} 1$$

Formula of the form

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

where $v_i \in V_a, P = \{a_1, \dots, a_n\}$ and $P \subseteq A$ will be called a P**-basic formula** or in short P**-formula**. A-basic formulas will be called **basic formulas**

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

Each row in the table is represented by a certain A-formula and the whole table is now represented by the set of all such formulas

Consider

Table 4: REE

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

$a_1b_0c_2, a_2b_0c_3, a_1b_1c_1, a_2b_1c_3, a_1b_0c_3$ are all basic formulas in the KR-system. The characteristic formula of the system is

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

Specific axioms of DL-logic

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. $\sim (a, v) \equiv \bigvee_{\substack{u \in V_a \\ u \neq v}} (a, u)$ for every $a \in A$

Proposition 7.3 (7.3)**.**

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

The axiom (1) follows from the assumption that each object can have exactly one value of each attribute.

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

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

The Proposition 7.3 means that the knowledge contained in the knowledge representation system is the whole knowledge available at the present stage, and corresponds to so called closed world assumption (CWA).

A formula ϕ is **derivable** from a set of formulas Ω, denoted $\Omega \vdash \phi$ if and only if it's derivable from axioms and formulas of Ω by finite application of modus ponens

Formula ϕ is a **theorem** of DL-logic, symbolically $\vdash \phi$ if it's derivable from the axioms only

A set of formulas Ω is **consistent** if and only if the formula $\phi \wedge \sim \phi$ is not derivable from Ω

7.4 Normal forms

Let $P \subseteq A$ and ϕ be a formula.

ϕ is in a P**-normal form** in S if and only if either ϕ is 0 or ϕ is 1, or ϕ is a disjunction of nonempty P-basic formulas in S

A-normal form will be referred to as **normal form**

Proposition 7.4 (7.4)**.** *Let ϕ be a formula in DL-language and let P contain all attributes occurring in ϕ. Moreover assume axioms (1)-(3) and the formulas $\sum_S(A)$. Then there is a formula ψ in the P-normal form s.t. $\vdash \phi \equiv \psi$*

7.5 Decision rules and decision algorithms

Any implication $\phi \to \psi$ will be called a **decision rule** in the KR-langauge. ϕ and ψ are referred to as the **predecessor** and the **successor** of $\phi \to \psi$ respectively.

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

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

If $\phi_1 \to \psi, \phi_2 \to \psi, \dots, \phi_n \to \psi$ are basic decision rules then the decision rule $\phi_1 \vee \phi_2 \vee \dots \vee \phi_n \to \psi$ will be called **combination** of basic decision rules $\phi_1 \to \psi, \phi_2 \to \psi, \dots, \phi_n \to \psi$ or in short **combined** decision rule.

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

Proposition 7.5 (7.5)**.** *A PQ-rule is true(consistent) if and only all*

$\{P \cup Q\}$ -basic formulas which occur in the $\{P \cup Q\}$ -normal form of the predecessor of the rule also occur in the $\{P \cup Q\}$ -normal form of the successor of the rule. Otherwise the rule is false

For example, the rule $b_0 \to c_2$ is false in 4, because 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

Any finite set of decision rules in a DL-language is referred to as a **decision algorithm** in the DL-language

Algorithm here means a set of instructions(decision rules)

Any finite set of basic decision rules will be called a **basic decision algorithm**.

If all decision rules in a basic decision algorithm are PQ-decision rules, then the algorithm is said to be PQ**-decision algoritbm**, 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 RP-rules admissible in S

A PQ-algorithm is **complete** in S if for every $x \in U$ there exists a PQ-decision rule $\phi \to \psi$ in the algorithm s.t. $x \vDash \phi \wedge \psi$ in S. Otherwise the algorithm is **incomplete**

The PQ-algorithm is **consistent** in S if and only if all its decision rules are consistent(true) in S. Otherwise **inconsistent**

Thus when we are given a KR-system, then any two arbitrary, nonempty subsets of attributes P, Q in the system, determine uniquely a PQ-decision algorithm and a decision table with P and Q as condition and decision attributes respectively. Hence a PQ-algorithm and PQ-decision table may be considered as equivalent concepts.

Consider

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

and assume $P = \{a, b, c\}$ and $Q = \{d, e\}$ are condition and decision attributes. Sets P and Q uniquely associate the following PQ-decision algorithm with the table:

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

7.6 Truth and indiscernibility

Proposition 7.6 (7.6)**.** *A PQ-decision rule $\phi \to \psi$ in a PQ-decision algorithm is consistent(true) in S if and only if for any PQ-decision rule $\phi' \to \psi'$ in PQ-decision algorithm, $\phi = \phi'$ implies $\psi = \psi'$*

Remark. in order to check whether or not a decision rule $\phi \to \psi$ 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.

7.7 Dependency of attributes

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, denoted by $P \Rightarrow_S Q$

It can be easily seen that the concept of dependency of attributes corresponds exactly to that introduced in CHAPTER 4

The set of attributes Q **depends partially** on the set of attributes P in S if there exists an inconsistent PQ-algorithm in S

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 PQ-rules in the algorithm.

In other words, the positive region of the decision algorithm (P, Q) is the consistent part of the inconsistent algorithm

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, we will say that the PQ-algorithm has the degree k

If a PQ-algorithm has degree k we can say that the set of attributes Q **depends in degree** k on the set of attributes P, and we will write $P \Rightarrow_k Q$

7.8 Reduction of consistent algorithms

Let (P, Q) be a consistent algorithm, and $a \in P$. Attribute a is **dispensable** in the algorithm (P, Q) if and only if the algorithm $((P - \{a\}), Q)$ is consistent. Otherwise **indispensable**

If all attributes $a \in P$ are dispensable in the algorithm (P, Q) then the algorithm (P, Q) will be called **independent**

The subset of attributes $R \subseteq P$ will be called a **reduct** of P in the algorithm (P, Q) if the algorithm (R, Q) is independent and consistent. (R, Q) is a **reduct** of (P, Q)

The set of all indispensable attributes in an algorithm (P, Q) will be called the **core** of the algorithm (P, Q), denoted by $CORE(P, Q)$

Proposition 7.7 (7.7)**.**

$$CORE(P, Q) = \bigcup RED(P, Q)$$

where $RED(P, Q)$ is the set of reducts of (P, Q)

Consider

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

and the PQ-algorithm in the system shown below

$$a_1 b_0 c_2 \to d_1 e_1$$
$$a_2 b_1 c_0 \to d_1 e_0$$
$$a_2 b_1 c_2 \to d_0 e_2$$
$$a_1 b_2 c_2 \to d_1 e_1$$
$$a_1 b_2 c_0 \to d_0 e_2$$

where $P = \{a, c, b\}$ and $Q = \{d, e\}$ are condition and decision attributes.

There are two reducts of P, namely $\{a, c\}$ and $\{b, c\}$

Note that

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

remove a we get

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

remove b we get

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

remove c we get

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

7.9 Reduction of inconsistent algorithms

Let (P, Q) be a inconsistent algorithm, and $a \in P$

An attribute a is **dispensable** in PQ-algorithm if $POS(P, Q) = POS((P-\{a\}), Q)$.

7.10 reduction of decision rules

If ϕ is P-basic formula and $Q \subseteq P$ then by ϕ/Q we mean the Q-basic formula obtained from the formula ϕ by removing from ϕ all elementary formulas (a, v_a) s.t. $a \in P - Q$

Let $\phi \to \psi$ be a PQ-rule and let $a \in P$. Attribute a is **dispensable** in the rule $\phi \to \psi$ if and only if

$$\models_S \phi \to \psi \text{ implies } \models \phi/(P - \{a\}) \to \psi$$

If all attributes $a \in P$ are dispensable in $\phi \to \psi$ then $\phi \to \psi$ will be called **independent**

The subset of attributes $R \subseteq P$ will be called a **reduct** of PQ-rule $\phi \to \psi$ if $\phi/R \to \psi$ is independent and $\models_S \phi \to \psi$ implies $\models_S \phi/R \to \psi$

If R is a reduct of the PQ-rule $\phi \to \psi$, then $\phi/R \to \psi$ is said to be **reduced**

The set of all indispensable attributes in $\phi \to \psi$ will be called the core of $\phi \to \psi$, and will be denoted by $CORE(\phi \to \psi)$.

Proposition 7.8 (7.8)**.**

$$CORE(P \to Q) = \bigcap RED(P \to Q)$$

There are two possibilities available at the moment. First we may reduce the algorithm, i.e. drop all dispensable condition attributes in the whole algorithm and afterwards reduce each decision rule in the reduced algorithm, i.e. drop all unnecessary conditions in each rule of the algorithm. The second option consists in reduction, at the very beginning, of decision rules, without the elimination of attributes from the whole algorithm

First

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

where $P = \{a, b, c\}$ and $Q = \{d, e\}$

$$\begin{aligned}
a_1 b_0 c_2 &\to d_1 e_1 \\
a_2 b_1 c_0 &\to d_1 e_0 \\
a_2 b_1 c_2 &\to d_0 e_2 \\
a_1 b_2 c_2 &\to d_1 e_1 \\
a_1 b_2 c_0 &\to d_0 e_2
\end{aligned}$$

For the first rule $a_1 b_0 c_2 \to d_1 e_1$, the core of it is the empty set. Either of the conditions $b_0 c_2, a_1 c_2, a_1 b_0$ uniquely determine the decision $d_1 e_1$. There are two reduct, namely $\{b\}$ and $\{a, c\}$

Below is the cores of each decision rule

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

And

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

Note that 1′ and 4 are identical.

7.11 Minimization of decision algorithm

Let $\mathbb{A}$ be a basic algorithm and let $S = (U, A)$ be a KR-system. The set of all basic rules in $\mathbb{A}$ having the same successor ψ will be denoted $\mathbb{A}_\psi$, and $\mathbb{P}_\psi$ is the set of all predecessors of decision rules belonging to $\mathbb{A}_\psi$

A basic decision rule $\phi \rightarrow \psi$ in $\mathbb{A}$ is **dispensable** in $\mathbb{A}$ if $\models_S \bigvee \mathbf{P}_\psi \equiv \bigvee \left\{ \mathbb{P}_\psi - \{\phi\} \right\}$. Otherwise **indispensable**

A subset $\mathbb{A}'$ of decision rules of $\mathbb{A}_\psi$ is a **reduct** of $\mathbb{A}_\psi$ if all decision rules in $\mathbb{A}'$ are independent and $\models_S \bigvee \mathbb{P}_\psi \equiv \bigvee \mathbb{P}'_\psi$

A set of decision rules $\mathbb{A}_\psi$ is **reduced** if reduct of $\mathbb{A}_\psi$ is $\mathbb{A}_\psi$ itself

A basic algorithm $\mathbb{A}$ is **minimal** if every decision rule in $\mathbb{A}$ is reduced and for every decision rule $\phi \rightarrow \psi$ in $\mathbb{A}$, $\mathbb{A}_\psi$ is reduced

Thus in order to simplify a PQ-algorithm, we must first reduce the set of attributes, i.e. we present the algorithm in a normal form (note that many normal forms are possible in general). The next step consists in the reduction of the algorithm, i.e. simplifying the decision rules. The last step removes all superfluous decision rules from the algorithm.

Given

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	2	0	2	2
7	2	2	2	2	2

and assume $P = \{a, b, c, d\}$ and $Q = \{e\}$ are condition and decision attributes

The only e-dispensable condition attribute is c

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	2	2	2
7	2	2	2	2

In the next step we have to reduce the superfluous values of attributes, i.e. reduce all decision rules in the algorithm. To this end we have first computed core values of attributes

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

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

Hence

$$a_1 b_0 \rightarrow e_1$$
$$a_0 \vee b_1 d_1 \rightarrow e_0$$
$$d_2 \rightarrow e_2$$