Rough Sets

(Granular Computing)

Basic Concepts of Rough Sets

- Information/Decision Systems (Tables)
- Indiscernibility
- Set Approximation
- Reducts and Core
- Dependency of Attributes

Information Systems/Tables

	Age	LEMS
x1	16-30	50
x2	16-30	0
x3	31-45	1-25
x4	31-45	1-25
x5	46-60	26-49
x6	16-30	26-49
x7	46-60	26-49

- IS is a pair (U, A)
- U is a non-empty finite set of objects.
- A is a non-empty finite set of attributes such that $a: U \rightarrow V_a$ for every $a \in A$.
- V_a is called the value set of a.

Decision Systems/Tables

	Age	LEMS	Walk
x1	16-30	50	yes
x2	16-30	0	no
x3	31-45	1-25	no
x4	31-45	1-25	yes
x5	46-60	26-49	no
x6	16-30	26-49	yes
x7	46-60	26-49	no

- DS: $T = (U, A \cup \{d\})$
- $d \notin A$ is the *decision* attribute (instead of one we can consider more decision attributes).
- The elements of A are called the *condition* attributes.

Indiscernibility

- The equivalence relation

  A binary relation $R \subseteq X \times X$ which is reflexive (xRx for any object x) ,

  symmetric (if xRy then yRx), and

  transitive (if xRy and yRz then xRz).

- The equivalence class $[x]_R$ of an element $x \in X$ consists of all objects $y \in X$ such that xRy.

Indiscernibility (2)

- Let *IS = (U, A)* be an information system, then with any $B \subseteq A$ there is an associated equivalence relation:

$$IND_{IS}(B) = \{(x, x') \in U^2 \mid \forall a \in B, a(x) = a(x')\}$$

where $IND_{IS}(B)$ is called the *B-indiscernibility relation.*

- If $(x, x') \in IND_{IS}(B)$, then objects *x* and *x'* are indiscernible from each other by attributes from *B*.
- The equivalence classes of the *B-indiscernibility relation* are denoted by $[x]_B$.

An Example of Indiscernibility

	Age	LEMS	Walk
x1	16-30	50	yes
x2	16-30	0	no
x3	31-45	1-25	no
x4	31-45	1-25	yes
x5	46-60	26-49	no
x6	16-30	26-49	yes
x7	46-60	26-49	no

- The non-empty subsets of the condition attributes are *{Age}*, *{LEMS}*, and *{Age, LEMS}*.
- *IND({Age}) = {{x1,x2,x6}, {x3,x4}, {x5,x7}}*
- *IND({LEMS}) = {{x1}, {x2}, {x3,x4}, {x5,x6,x7}}*
- *IND({Age,LEMS}) = {{x1}, {x2}, {x3,x4}, {x5,x7}, {x6}}*.

Set Approximation

- Let $T = (U, A)$ and let $B \subseteq A$ and $X \subseteq U$. We can approximate X using only the information contained in B by constructing the B-*lower* and B-*upper* approximations of X, denoted $\underline{B}X$ and $\overline{B}X$ respectively, where

$$\underline{B}X = \{x \mid [x]_B \subseteq X\},$$

$$\overline{B}X = \{x \mid [x]_B \cap X \neq \phi\}.$$

An Example of Set Approximation

	Age	LEMS	Walk
x1	16-30	50	yes
x2	16-30	0	no
x3	31-45	1-25	no
x4	31-45	1-25	yes
x5	46-60	26-49	no
x6	16-30	26-49	yes
x7	46-60	26-49	no

- Let W = {x | Walk(x) = yes}.

$$\underline{A}W = \{x1, x6\},$$

$$\overline{A}W = \{x1, x3, x4, x6\},$$

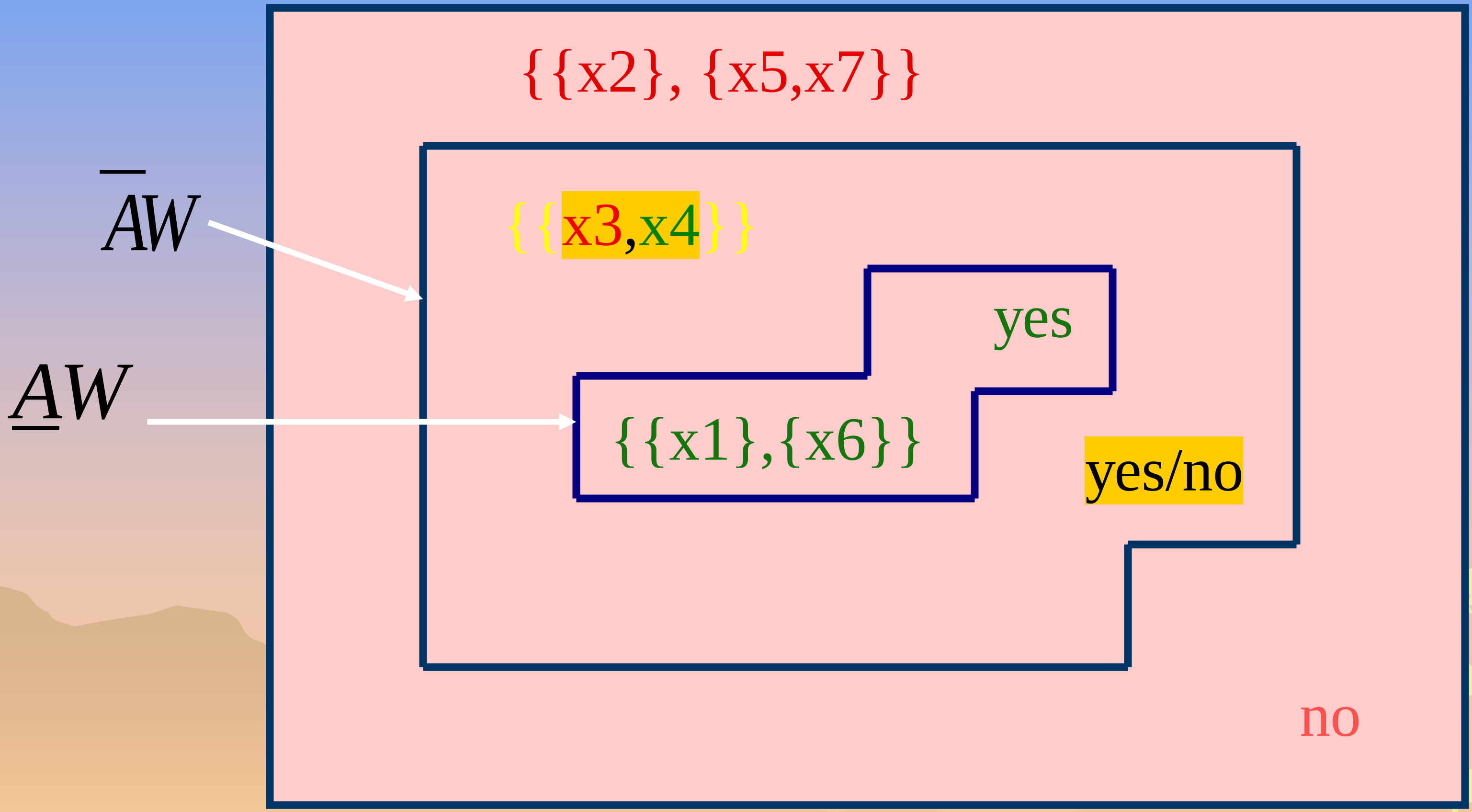
$$BN_A(W) = \{x3, x4\},$$

$$U - \overline{A}W = \{x2, x5, x7\}.$$

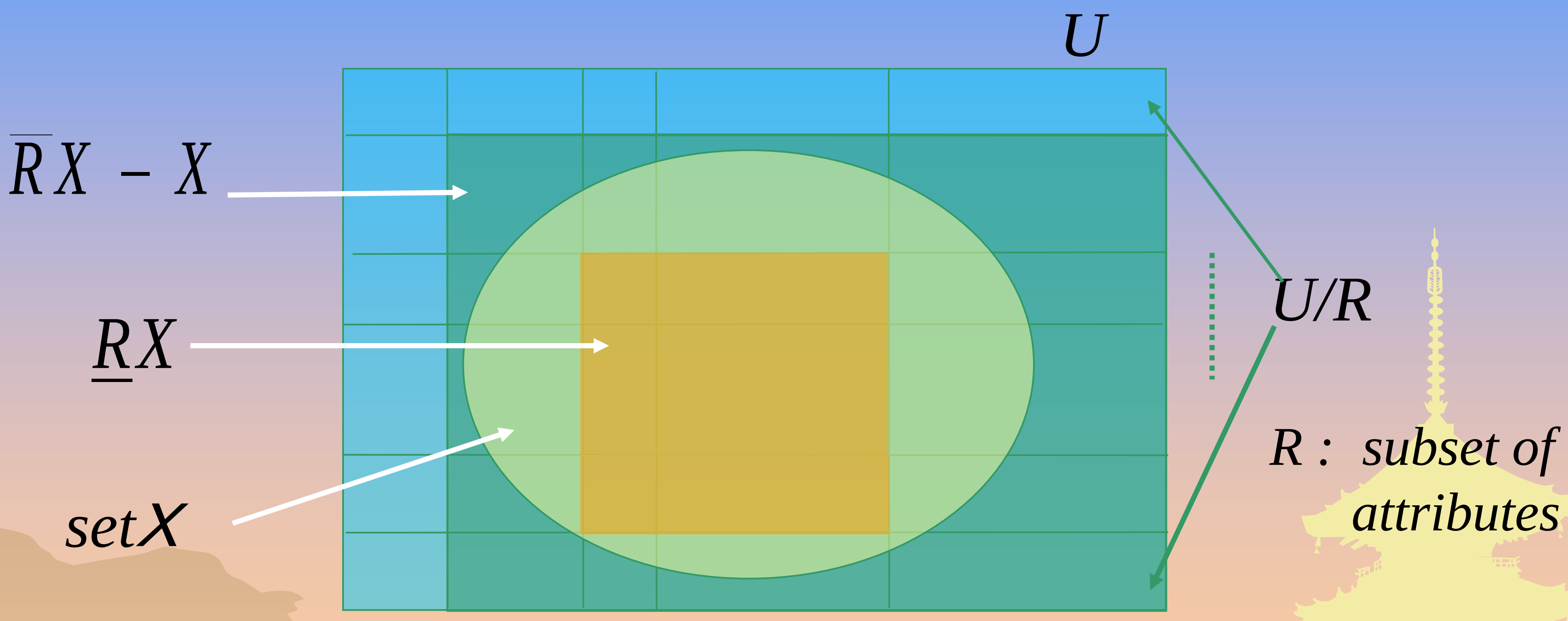
- The decision class, *Walk*, is rough since the boundary region is not empty.

IND({Age, LEMS}) = {{x1}, {x2}, {x3,x4}, {x5,x7}, {x6}}

An Example of Set Approximation (2)



Lower & Upper Approximations



Lower & Upper Approximations (2)

Upper Approximation:

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

Lower Approximation:

$$\underline{R}X = \bigcup \{Y \in U / R : Y \subseteq X\}$$

Lower & Upper Approximations (3)

U	Headache	Temp.	Flu
U1	Yes	Normal	No
U2	Yes	High	Yes
U3	Yes	Very-high	Yes
U4	No	Normal	No
U5	No	High	No
U6	No	Very-high	Yes
U7	No	High	Yes
U8	No	Very-high	No

The indiscernibility classes defined by *R = {Headache, Temp.}* are
{u1}, {u2}, {u3}, {u4}, {u5, u7}, {u6, u8}.

X1 = {u | Flu(u) = yes}

= *{u2, u3, u6, u7}*

$\underline{R}X1$ = *{u2, u3}*

$\overline{R}X1$ = *{u2, u3, u6, u7,* ***u8, u5****}*

X2 = {u | Flu(u) = no}

= *{u1, u4, u5, u8}*

$\underline{R}X2$ = *{u1, u4}*

$\overline{R}X2$ = *{u1, u4, u5, u8,* ***u7, u6****}*

Lower & Upper Approximations (4)

R = {Headache, Temp.}

U/R = { *{u1}, {u2}, {u3}, {u4}, {u5, u7}, {u6, u8}*}

X1 = {u | Flu(u) = yes} = {*u2,u3,u6,u7*}

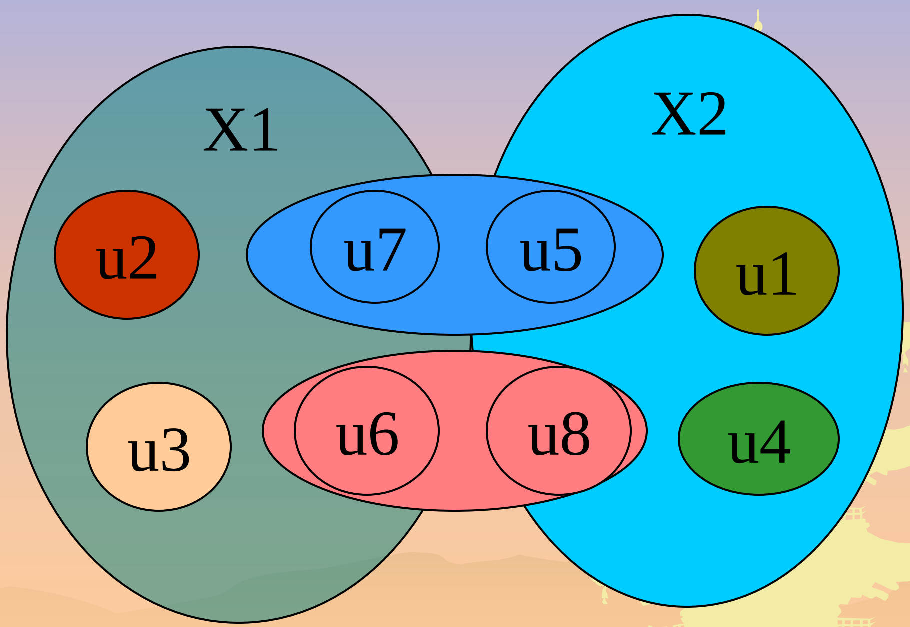
X2 = {u | Flu(u) = no} = {*u1,u4,u5,u8*}

$\underline{R}X1 = \{u2, u3\}$

$\overline{R}X1 = \{u2, u3, u6, u7, \mathbf{u8, u5}\}$

$\underline{R}X2 = \{u1, u4\}$

$\overline{R}X2 = \{u1, u4, u5, u8, \mathbf{u7, u6}\}$



Issues in the Decision Table

- The same or indiscernible objects may be represented several times.
- ***Some of the attributes may be superfluous (redundant).***

  That is, their removal cannot worsen the classification.

Reducts

- Keep only those attributes that preserve the indiscernibility relation and, consequently, set approximation.
- There are usually several such subsets of attributes and those which are minimal are called *reducts*.

Reduct & Core

- The set of attributes $R \subseteq C$ is called a *reduct* of *C*, if *T' = (U, R, D)* is independent and $POS_R(D) = POS_C(D)$.
- The set of all the condition attributes indispensable in *T* is denoted by *CORE(C)*.

$$CORE(C) = \cap RED(C)$$

where *RED(C)* is the set of all *reducts* of *C*.

An Example of Reducts & Core

U	*Headache*	*Muscle pain*	*Temp.*	*Flu*
U1	Yes	Yes	Normal	No
U2	Yes	Yes	High	Yes
U3	Yes	Yes	Very-high	Yes
U4	No	Yes	Normal	No
U5	No	No	High	No
U6	No	Yes	Very-high	Yes

Reduct1 **= {Muscle-pain,Temp.}**

U	*Muscle pain*	*Temp.*	*Flu*
U1,U4	Yes	Normal	No
U2	Yes	High	Yes
U3,U6	Yes	Very-high	Yes
U5	No	High	No

Reduct2 **= {Headache, Temp.}**

U	*Headache*	*Temp.*	*Flu*
U1	Yes	Norlmal	No
U2	Yes	High	Yes
U3	Yes	Very-high	Yes
U4	No	Normal	No
U5	No	High	No
U6	No	Very-high	Yes

CORE **= {Headache,Temp} ∩ {MusclePain, Temp} = {Temp}**

Discernibility Matrix
(used to find reducts)

- Let *T = (U, C, D)* be a decision table, with $U = \{u_1, u_2, ..., u_n\}$.

  By a discernibility matrix of *T*, denoted *M(T)*, we will mean $n \times n$ matrix defined as:

$$m_{ij} = \begin{cases} \{c \in C: c(u_i) \neq c(u_j)\} & if\ \exists d \in D[d(u_i) \neq d(u_j)] \\ \lambda & if\ \forall d \in D[d(u_i) = d(u_j)] \end{cases}$$

  for i, j = 1,2,…,n such that u_i or u_j belongs to the *C*-positive region of *D.*

- m_{ij} is the set of all the condition attributes that classify objects *ui* and *uj* into different classes.

Discernibility Function

- For any $u_i \in U$,

$$f_T(u_i) = \wedge_j \{\vee m_{ij} : j \neq i, j \in \{1,2,...,n\}\}$$

where (1) $\vee m_{ij}$ is the disjunction of all variables a such that $a \in m_{ij}$, if $m_{ij} \neq \phi$.

(2) $\vee m_{ij} = \bot(false)$, if $m_{ij} = \phi$.

(3) $\vee m_{ij} = t(true)$, if $m_{ij} = \lambda$.

Each logical product in the minimal disjunctive normal form (DNF) defines a reduct of instance u_i.

Example of Discernibility Matrix

No	a	b	c	d
u1	*a0*	*b1*	*c1*	*y*
u2	*a1*	*b1*	*c0*	*n*
u3	*a0*	*b2*	*c1*	*n*
u4	*a1*	*b1*	*c1*	*y*

$C = \{a, b, c\}$

$D = \{d\}$

$$(a \vee c) \wedge b \wedge c \wedge (a \vee b)$$

$$= b \wedge c$$

Reduct = $\{b, c\}$

In order to discern equivalence classes of the decision attribute *d*, to preserve conditions described by the discernibility matrix for this table

	u1	u2	u3
u2	a,c		
u3	b	λ	
u4	λ	c	a,b

Example of Discernibility Matrix (2)

	a	b	c	d	E
u1	1	0	2	1	1
u2	1	0	2	0	1
u3	1	2	0	0	2
u4	1	2	2	1	0
u5	2	1	0	0	2
u6	2	1	1	0	2
u7	2	1	2	1	1

	u1	u2	u3	u4	u5	u6
u2	λ					
u3	b,c,d	b,c				
u4	b	b,d	c,d			
u5	a,b,c,d	a,b,c	λ	a,b,c,d		
u6	a,b,c,d	a,b,c	λ	a,b,c,d	λ	
u7	λ	λ	a,b,c,d	a,b	c,d	c,d

Core = {b}

Reduct1 = {b,c}

Reduct2 = {b,d}

F(a,b,c,d)=(b+c+d)b(a+b+c+d)(b+c)(b+d)(a+b+c)(c+d)(a+b)=
b(c+d)=bc+bd

Reducts: {b,c}, {b,d}

The Goal of Attribute Selection

Finding an optimal subset of attributes in a database according to some criterion, so that a classifier with the highest possible accuracy can be induced by learning algorithm using information about data available only from the subset of attributes.

Attribute Evaluation Criteria

- Selecting the attributes that cause the number of consistent instances to increase faster
  - To obtain the subset of attributes as small as possible
- Selecting an attribute that has smaller number of different values
  - To guarantee that the number of instances covered by rules is as large as possible.

An Example of Attribute Selection

U	*a*	*b*	*c*	*d*	*e*
u1	1	0	2	1	1
u2	1	0	2	0	1
u3	1	2	0	0	2
u4	1	2	2	1	0
u5	2	1	0	0	2
u6	2	1	1	0	2
u7	2	1	2	1	1

Condition Attributes:

a: $Va = \{1, 2\}$

b: $Vb = \{0, 1, 2\}$

c: $Vc = \{0, 1, 2\}$

d: $Vd = \{0, 1\}$

Decision Attribute:

e: $Ve = \{0, 1, 2\}$

Searching for *CORE*

Removing attribute *a*

U	*b*	*c*	*d*	*e*
u1	0	2	1	1
u2	0	2	0	1
u3	2	0	0	2
u4	2	2	1	0
u5	1	0	0	2
u6	1	1	0	2
u7	1	2	1	1

Removing attribute *a* does not cause inconsistency.

Hence, *a* is not used as *CORE*.

Searching for *CORE* (2)

Removing attribute *b*

U	*a*	*c*	*d*	*e*
u1	1	2	1	1
u2	1	2	0	1
u3	1	0	0	2
u4	1	2	1	0
u5	2	0	0	2
u6	2	1	0	2
u7	2	2	1	1

Removing attribute *b* cause inconsistency.

$$u_1 : a_1 c_2 d_1 \rightarrow e_1$$

$$u_4 : a_1 c_2 d_1 \rightarrow e_0$$

Hence, *b* is used as CORE.

Searching for *CORE* (3)

Removing attribute *c*

U	*a*	*b*	*d*	*e*
u1	1	0	1	1
u2	1	0	0	1
u3	1	2	0	2
u4	1	2	1	0
u5	2	1	0	2
u6	2	1	0	2
u7	2	1	1	1

Removing attribute *c* does not cause inconsistency.

Hence, *c* is not used as *CORE*.

Searching for *CORE* (4)

Removing attribute *d*

U	*a*	*b*	*c*	*e*
u1	1	0	2	1
u2	1	0	2	1
u3	1	2	0	2
u4	1	2	2	0
u5	2	1	0	2
u6	2	1	1	2
u7	2	1	2	1

Removing attribute *d* does not cause inconsistency.

Hence, *d* is not used as *CORE*.

Searching for *CORE* (5)

Attribute *b* is the unique indispensable attribute.

$$CORE(C)=\{b\}$$

Initial subset $R = \{b\}$

$R=\{b\}$

T

U	a	*b*	*c*	*d*	*e*
u1	1	0	2	1	1
u2	1	0	2	0	1
u3	1	2	0	0	2
u4	1	2	2	1	0
u5	2	1	0	0	2
u6	2	1	1	0	2
u7	2	1	2	1	1

T'

U'	*b*	*e*
u1	0	1
u2	0	1
u3	2	2
u4	2	0
u5	1	2
u6	1	2
u7	1	1

$\because b_0 \longrightarrow e_1$

The instances containing *b0* will not be considered.

Attribute Evaluation Criteria

- Selecting the attributes that cause the number of consistent instances to increase faster
  - To obtain the subset of attributes as small as possible
- Selecting the attribute that has smaller number of different values
  - To guarantee that the number of instances covered by a rule is as large as possible.

Selecting Attribute from $\{a,c,d\}$

1. Selecting $\{a\}$

$R = \{a,b\}$

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

$a1b2 \rightarrow e2$

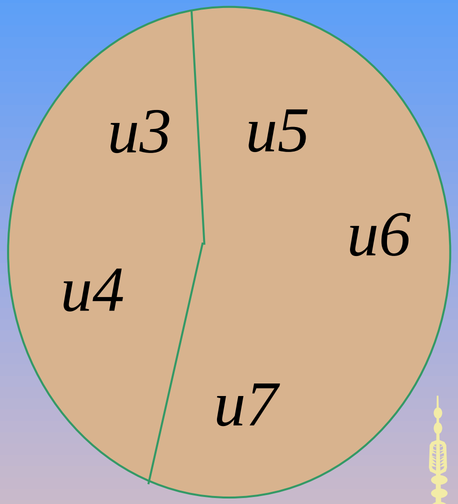
$a1b2 \rightarrow e0$

$a2b1 \rightarrow e2$

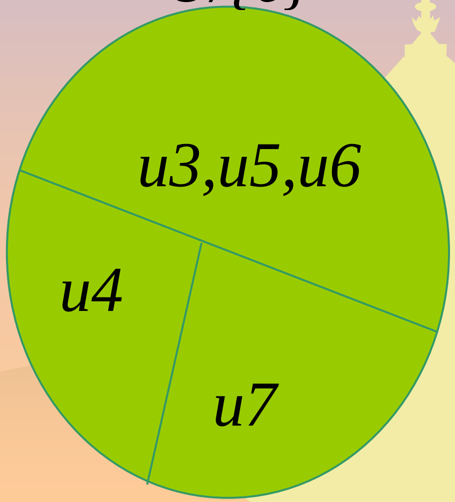
$a2b1 \rightarrow e1$

$$\bigcup_{X \in U/\{e\}} POS_{\{a,b\}}(X) = \phi$$

$U/\{a,b\}$



$U/\{e\}$



Selecting Attribute from {*a*,*c*,*d*} (2)

2. Selecting {*c*}

$R = \{b,c\}$

U'	*b*	*c*	*e*
u3	2	0	2
u4	2	2	0
u5	1	0	2
u6	1	1	2
u7	1	2	1

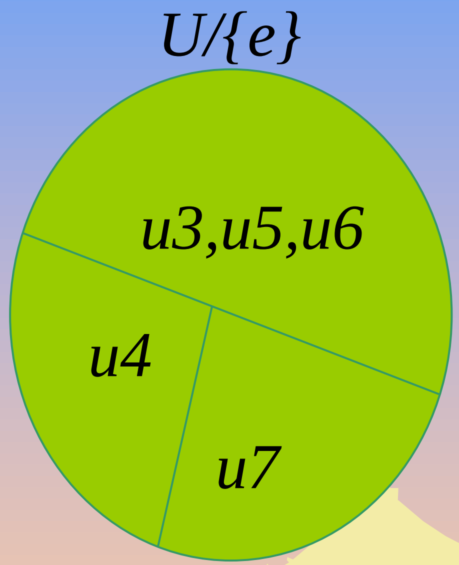
$$b_2c_0 \longrightarrow e_2$$

$$b_2c_2 \longrightarrow e_0$$

$$b_1c_0 \longrightarrow e_2$$

$$b_1c_1 \longrightarrow e_2$$

$$b_1c_2 \longrightarrow e_1$$



$$\bigcup_{X \in U/\{e\}} POS_{\{b,c\}}(X) = \{u3, u4, u5, u6, u7\};$$

Selecting Attribute from {*a*,*c*,*d*} (3)

3. Selecting {*d*}

$R = \{b,d\}$

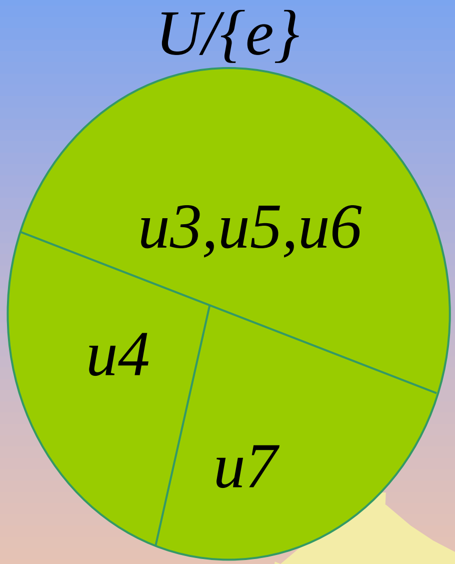
U'	b	d	e
u3	2	0	2
u4	2	1	0
u5	1	0	2
u6	1	0	2
u7	1	1	1

$b_2 d_0 \rightarrow e_2$

$b_2 d_1 \rightarrow e_0$

$b_1 d_0 \rightarrow e_2$

$b_1 d_1 \rightarrow e_1$

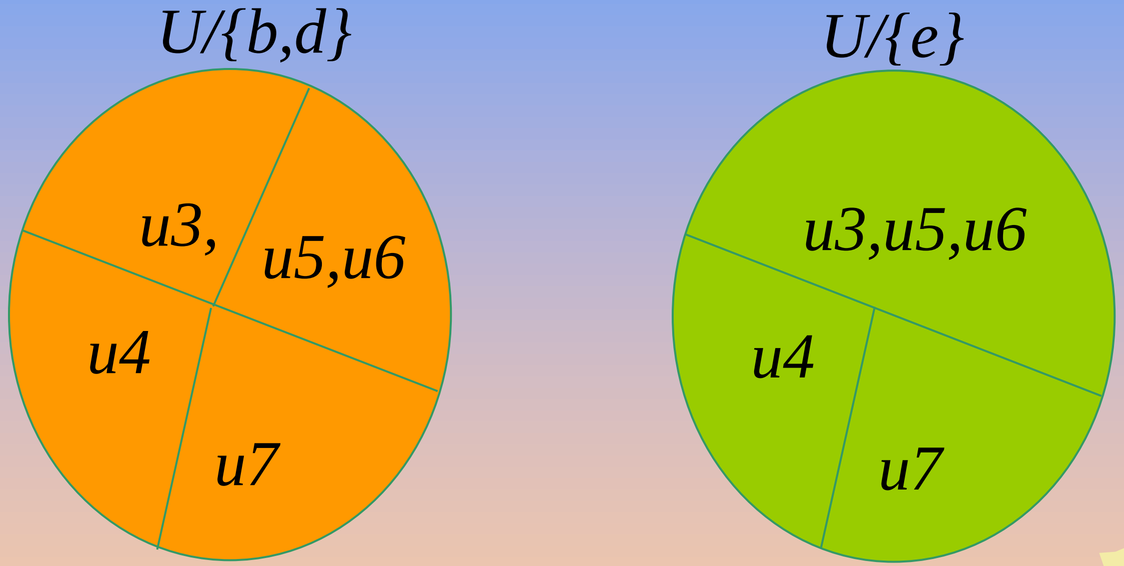


$$\bigcup_{X \in U/\{e\}} POS_{\{b,d\}}(X) = \{u3, u4, u5, u6, u7\};$$

Selecting Attribute from *{a,c,d}* (4)

3. Selecting *{d}*

$R = \{b,d\}$



$$POS_{\{b,d\}}(\{u3,u5,u6\})/\{b,d\} = \{\{u3\},\underline{\{u5,u6\}}\}$$

$$\max\_size(POS_{\{b,d\}}(\{u3,u5,u6\})/\{b,d\}) = 2$$

Result: Subset of attributes = {b, d}