Finish Fuzzy Sets and Logic

Begin Rough Sets

## Why Use Fuzzy Logic (FL)?

1) FL offers several unique features that make it a particularly good choice for many control problems.
2) FL is inherently robust since it does not require precise, noise-free inputs and can be programmed to fail safely if a feedback sensor quits or is destroyed. The output control is a smooth control function despite a wide range of input variations.
3) Since the FL controller processes user-defined rules governing the target control system, it can be modified and tweaked easily to improve or drastically alter system performance. New sensors can easily be incorporated into the system simply by generating appropriate governing rules.



## Why Use Fuzzy Logic? (cont)

4) FL is not limited to a few feedback inputs and one or two control outputs, nor is it necessary to measure or compute rate-of-change parameters in order for it to be implemented. Any sensor data that provides some indication of a system's actions and reactions is sufficient. This allows the sensors to be inexpensive and imprecise thus keeping the overall system cost and complexity low.

5) Because of the rule-based operation, any reasonable number of inputs can be processed and numerous outputs generated, although defining the rule base quickly becomes complex if too many inputs and outputs are chosen for a single implementation since rules defining their interrelations must also be defined. It would be better to divide the control system into smaller chunks and use several smaller FL controllers distributed on the system.



## Why Use Fuzzy Logic? (cont)

6) FL can control nonlinear systems that would be difficult or impossible to model mathematically. This opens doors for control systems that would normally be deemed unfeasible for automation.



## How does Fuzzy Logic Work?

FL requires numerical parameters to operate such as what is considered significant error and significant rate-of-change-of-error, but exact values of these numbers are usually not critical unless responsive performance is required in which case empirical tuning would determine them. E.g., a simple temperature control system could use a single temperature feedback sensor whose data is subtracted from the command signal to compute "error" and then time-differentiated to yield the rate-of-change-of-error, called "error-dot". Error might have units of degs F and a small error considered to be 2F while a large error is 5F. The "error-dot" might then have units of degs/min with a small error-dot being 5F/min and a large one being 15F/min. These values don't have to be symmetrical and can be "tweaked" once the system is operating in order to optimize performance. Generally, FL is so forgiving that the system will probably work the first time without any tweaking.

## How is Fuzzy Logic Used?

1) Define the control objectives and criteria: What am I trying to control? What do I have to do to control the system? What kind of response do I need? What are the possible (probable) system failure modes?
2) Determine the input and output relationships and choose a minimum number of variables for input to the FL engine (typically error and rate-of-change-of-error).

## How is Fuzzy Logic Used? (cont)

3) Using the rule-based structure of FL, break the control problem down into a series of IF X AND Y THEN Z rules that define the desired system output response for given system input conditions. The number and complexity of rules depends on the number of input parameters that are to be processed and the number fuzzy variables associated with each parameter. If possible, use at least one variable and its time derivative. Although it is possible to use a single, instantaneous error parameter without knowing its rate of change, this cripples the system's ability to minimize overshoot for a step inputs.

4) Create FL membership functions that define the meaning (values) of Input/Output terms used in the rules.



## How is Fuzzy Logic Used? (cont)

5) Create the necessary pre- and post-processing FL routines if implementing in S/W, otherwise program the rules into the FL H/W engine.

6) Test the system, evaluate the results, tune the rules and membership functions, and retest until satisfactory results are obtained.




## Linguistic Variables

In 1973, Professor Lotfi Zadeh proposed the concept of linguistic or "fuzzy" variables. Think of them as linguistic objects or words, rather than numbers. The sensor input is a noun, e.g. "temperature", "displacement", "velocity", "flow", "pressure", etc. Since error is just the difference, it can be thought of the same way. The fuzzy variables themselves are adjectives that modify the variable (e.g. "large positive" error, "small positive" error ,"zero" error, "small negative" error, and "large negative" error). As a minimum, one could simply have "positive", "zero", and "negative" variables for each of the parameters. Additional ranges such as "very large" and "very small" could also be added to extend the responsiveness to exceptional or very nonlinear conditions, but aren't necessary in a basic system.



## Examples

The following five cases show what the system computes as error decreases toward zero and then changes to a positive value. Pictures of the input and output membership functions are included. The rate-of-change of the error stays constant throughout the five cases. It is not likely that this would happen in a real system, but for purposes of this illustration, that case has been assumed. The values of error and error-dot indicated from the membership functions are plugged into the rulebase from the "KEY" below and the responses computed for each case. These responses are then mathematically combined to yield a crisp output.



## Examples (cont)

Note that because the "zero" membership function is centered on zero in the output function, its influence in the output computation is only in the denominator. The center of the "zero" doesn't need to be at zero, it just happens to be in this example.

KEY:

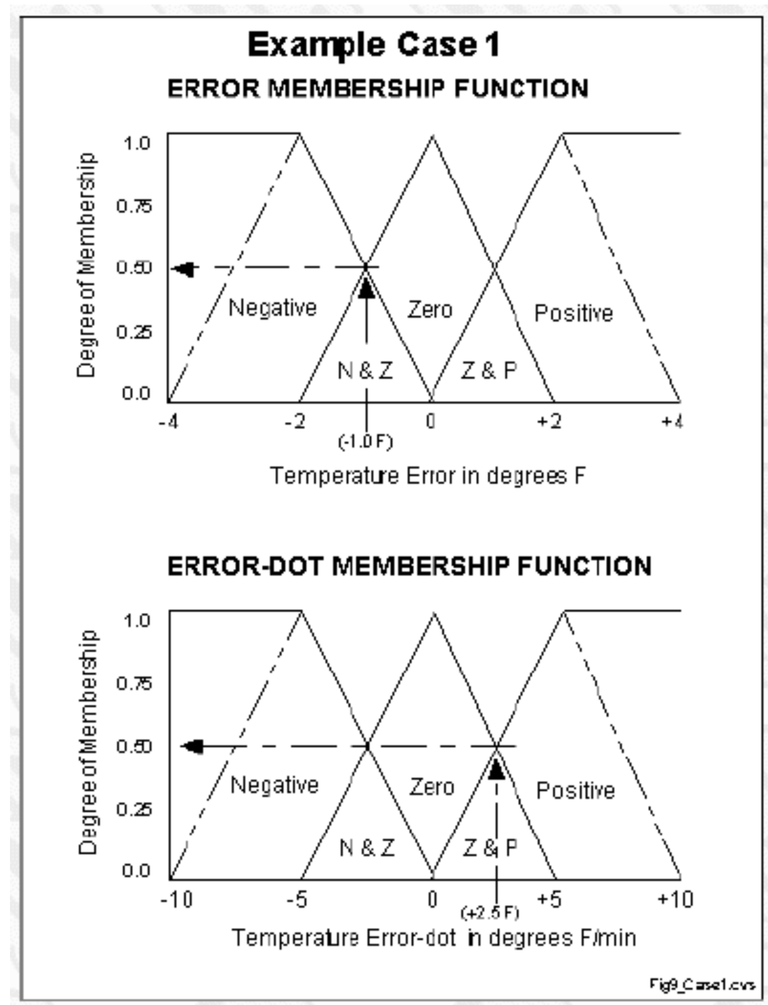
(e<0) "negative" error value (er<0) "negative" error-dot value

(e=0) "zero" error value (er=0) "zero" error-dot value

(e>0) "positive" error value (er>0) "positive" error-dot value

## Examples (cont)

Case 1 – error = 1.0F
(initial conditions)



## Examples (cont)

Input degree of membership

"error" = -1.0: "negative" = 0.5, "zero" = 0.5, "positive" = 0.0

"error-dot" = +2.5: "negative" = 0.0, "zero" = 0.5, "positive" = 0.5

1. If (e < 0) AND (er < 0) then Cool 0.50 & 0.00 = 0.00
2. If (e = 0) AND (er < 0) then Heat 0.50 & 0.00 = 0.00
3. If (e > 0) AND (er < 0) then Heat 0.00 & 0.00 = 0.00
4. If (e < 0) AND (er = 0) then Cool 0.50 & 0.50 = 0.50
5. If (e = 0) AND (er = 0) then No_Chng 0.50 & 0.50 = 0.50
6. If (e > 0) AND (er = 0) then Heat 0.00 & 0.50 = 0.00
7. If (e < 0) AND (er > 0) then Cool 0.50 & 0.50 = 0.50
8. If (e = 0) AND (er > 0) then Cool 0.50 & 0.50 = 0.50
9. If (e > 0) AND (er > 0) then Heat 0.00 & 0.50 = 0.00



## Examples (cont)

"negative" = $(R1^2 + R4^2 + R7^2 + R8^2)$ (Cooling) = $(0.00^2 + 0.50^2 + 0.50^2 + 0.50^2)^{.5} = 0.866$

"zero" = $(R5^2)^{.5} = (0.50^2)^{.5}$ (No Change) = $0.500$

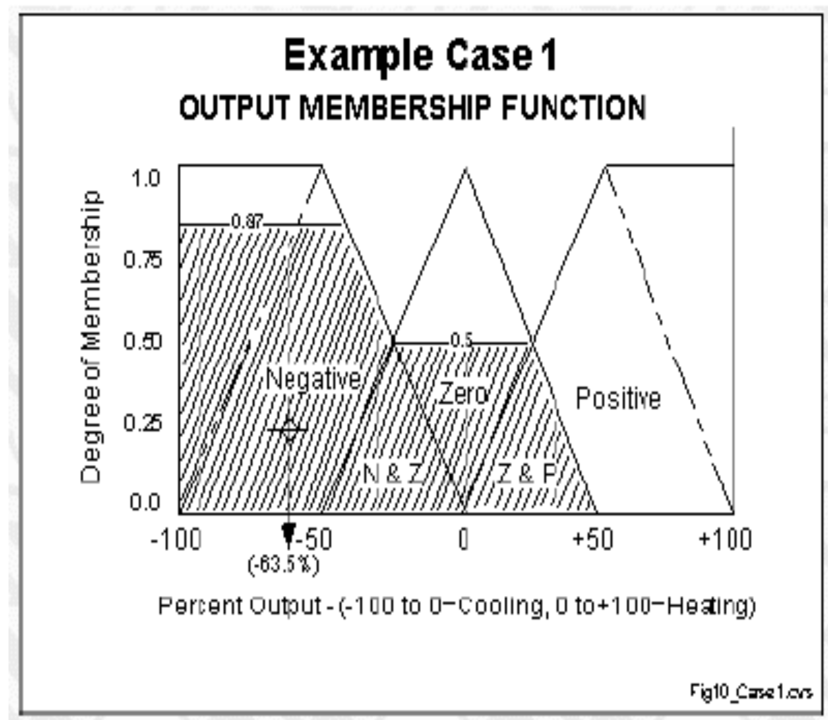
"positive" = $(R2^2 + R3^2 + R6^2 + R9^2)$ (Heating) = $(0.00^2 + 0.00^2 + 0.00^2 + 0.00^2)^{.5} = 0.000$

## Examples (cont)

$$\frac{[(neg\_cntr \times neg\_strength + zero\_cntr \times zero\_strength + pos\_cntr \times pos\_strength)]}{[(neg\_strength + zero\_strength + pos\_strength)]} = Output$$

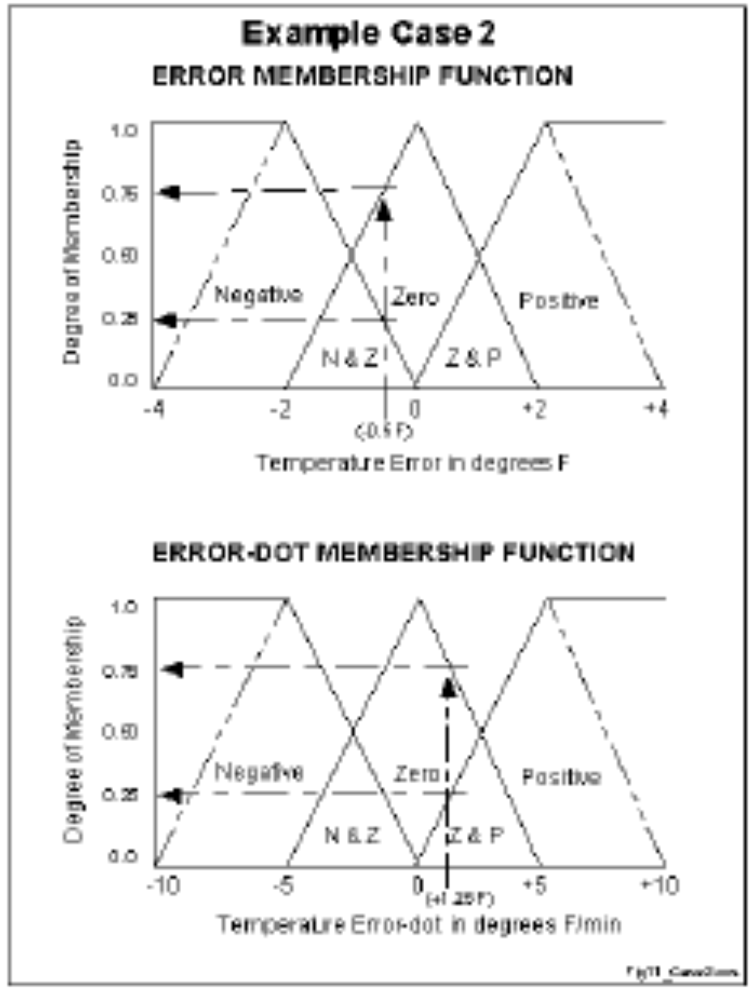
$$\frac{(-100 \times 0.866 + \times 0.500 + 100 \times 0.000)}{(0.866 + 0.500 + 0.000)}$$

= -63.4% (cooling



## Examples (cont)

Case 2 – error = +1.25F





## Examples (cont)

Input Degree of Membership

"error" = -0.5: "negative" = 0.25, "zero" = 0.75, "positive" = 0.0

"error-dot" = +2.5: "negative" = 0.0, "zero" = 0.50, "positive" = 0.50

1. If (e < 0) AND (er < 0) then Cool 0.25 & 0.0 = 0.00
2. If (e = 0) AND (er < 0) then Heat 0.75 & 0.0 = 0.00
3. If (e > 0) AND (er < 0) then Heat 0.00 & 0.0 = 0.00
4. If (e < 0) AND (er = 0) then Cool 0.25 & 0.50 = 0.25
5. If (e = 0) AND (er = 0) then No_Chng 0.75 & 0.50 = 0.50
6. If (e > 0) AND (er = 0) then Heat 0.00 & 0.50 = 0.00
7. If (e < 0) AND (er > 0) then Cool 0.25 & 0.50 = 0.25
8. If (e = 0) AND (er > 0) then Cool 0.75 & 0.50 = 0.50
9. If (e > 0) AND (er > 0) then Heat 0.00 & 0.50 = 0.00



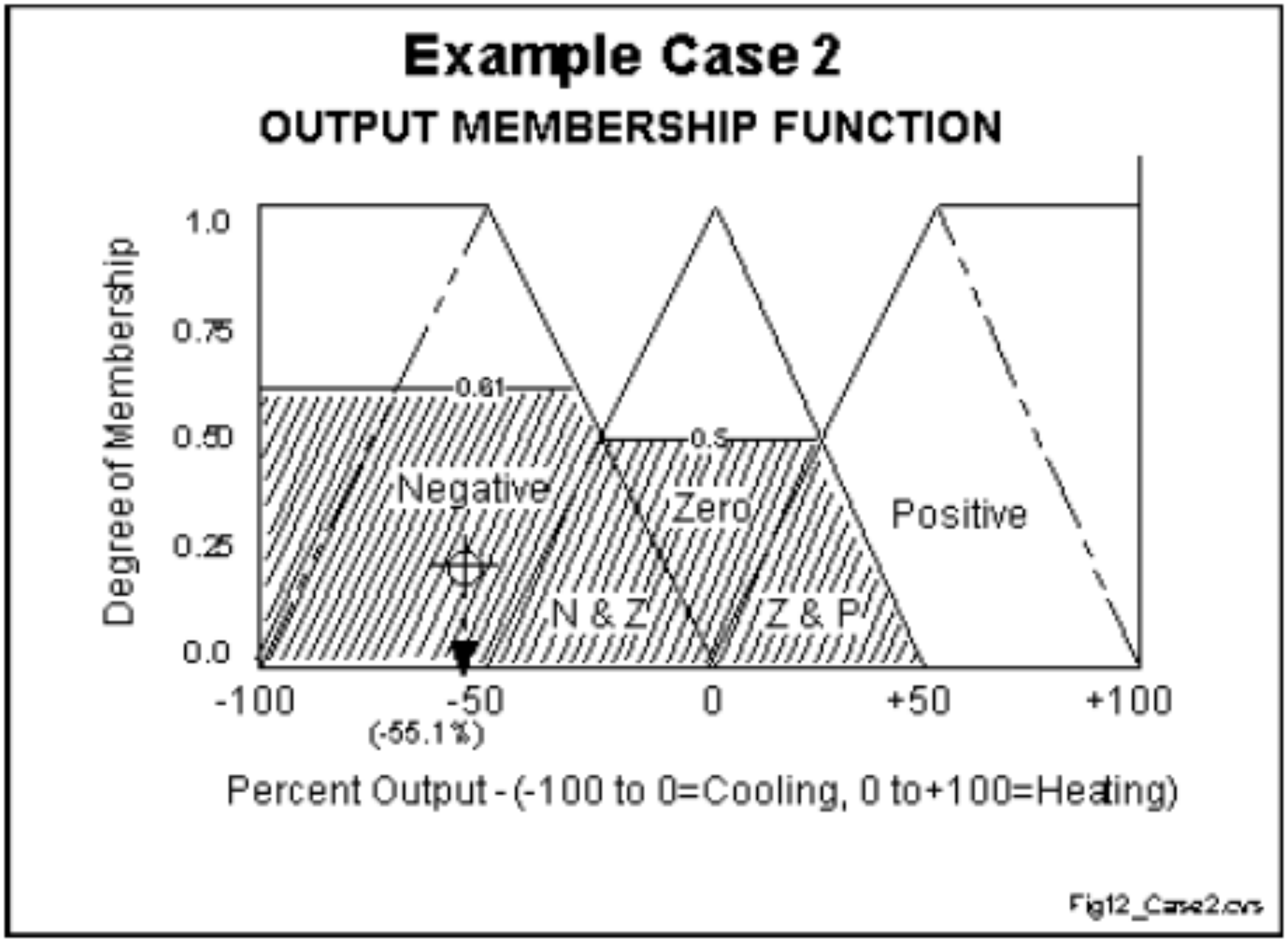

## Examples (cont)

"negative" = (R1^2 + R4^2 + R7^2 + R8^2) (Cooling) = (0.00^2 + 0.25^2 + 0.25^2 + 0.50^2)^.5 = 0.612

"zero" = (R5^2)^.5 = (0.50^2)^.5 (No Change) = 0.50

"positive" = (R2^2 + R3^2 + R6^2 + R9^2) (Heating) = (0.00^2 + 0.00^2 + 0.00^2 + 0.00^2)^.5 = 0.000

# Examples (cont)





## Examples (cont)

[(neg_cntr×neg_strength+zero_cntrr×zero_strength
+pos_cntr×pos_strength) / [((neg_strength + zero_strength + pos_strength)]

= OUTPUT

(-100 × 0.612 + 0 × 0.50 + 100 * 0.000) = -55.1% (cooling)
(0.612 + 0.500 + 0.000)

# Examples (cont)

And so on (look at handout)



## Rough Sets

Rough set theory was introduced by Zdzislaw Pawlak in 1982

Representative Publications:

- Z. Pawlak, Rough Sets, International Journal of Computer and Information Sciences, Vol.11, 341-356 (1982).
- Z. Pawlak, Rough Sets - Theoretical Aspect of Reasoning about Data, Kluwer Academic Pubilishers (1991).

Rough Sets is useful for reasoning about knowledge of objects represented by attributes (features).

Fundamental Assumptions:

- Objects are represented by values of attributes.
- Objects with the same information are indiscernible.



## Rough Sets - Basic Concepts

Approximation Space An approximation space is a pair $(U, R)$ where $U$ is a nonempty finite set called the universe and $R$ is an equivalence relation defined on $U$.

Information System An information 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 attributes, i.e., $a: U \rightarrow V_a$ for $a \in A$, where $V_a$ is called the domain of $a$.

Decision Table (Data Table) A decision table is a special case of information systems, $S = (U, A = C \cup \{d\})$,

where attributes in $C$ are called condition attributes and $d$ is a designated attribute called the decision attribute.



## Rough Sets

### Table 1 - Example of a Decision Table

| STUDENT | Category | Major | Birth_Place | Grade |
|---|---|---|---|---|
| 1 | PhD | History | Detroit | A |
| 2 | MS | Chemistry | Akron | A |
| 3 | MS | History | Detroit | C |
| 4 | BS | Math | Detroit | B |
| 5 | BS | Chemistry | Akron | C |
| 6 | PhD | Computing | Cleveland | A |
| 7 | BS | Chemistry | Cleveland | C |
| 8 | PhD | Computing | Akron | A |

## Rough Sets

### Approximation of Sets

Let S = (U, R) be an approximation space and X be a subset of U.

The lower approximation of X by R in S is defined as

$$\underline{R}X = \{ e \in U \mid [e] \subseteq X \} \text{ and}$$

The upper approximation of X by R in S is defined as

$$\overline{R}X = \{ e \in U \mid [e] \cap X \neq \Theta \}$$

where [e] denotes the equivalence class containing e. [e] is called elementary set.



## Rough Sets

A subset X of U is said to be R-definable in S if and only if $\underline{R}X = \overline{R}X$

The boundary set $BN_R(X)$ is defined as $\overline{R}X - \underline{R}X$

A set is rough in S if its boundary set is nonempty.

## Rough Sets - Accuracy of Approximations

$$\alpha_B(X) = \frac{|\underline{B}(X)|}{|\overline{B}(X)|}$$

where S = (U, A), $B \subseteq A$ and $X \subseteq U$

|X| denotes the cardinality of X

If $\alpha_B(X) = 1$ then X is crisp with respect to B

If $\alpha_B(X) < 1$ then X is rough with respect to B

## Rough Sets

| U | Category | Major | Birth_Place | Grade |
|---|---|---|---|---|
| 1 | PhD | History | Detroit | A |
| 2 | MS | Chemistry | Akron | A |
| 3 | MS | History | Detroit | C |
| 4 | BS | Math | Detroit | B |
| 5 | BS | Chemistry | Akron | C |
| 6 | PhD | Computing | Cleveland | A |
| 7 | BS | Chemistry | Cleveland | C |
| 8 | PhD | Computing | Akron | A |

U\{Category} = {{1, 6, 8}, {2, 3}, {4, 5, 7}}
U\{Major} = {{1, 3}, {2, 5, 7}, {4}, {6, 8}}
U\{Birth_Place} = {{2, 5, 8}, {1, 3, 4}, {6, 7}}
U\{Grade} = {{1, 2, 6, 8}, {4}, {3, 5, 7}}

Accuracy:
$\alpha_B(X)=2/6=1/3$

Let X = [(Grade, A)] = {1, 2, 6, 8}
Let B = {Major, Birth_PLACE}
U\B = {{1, 3}, {2, 5}, {4}, {6}, {7}, {8}}

$\underline{B}(X)$ = BX
B(X) = {1,2,3,5,6,8}
$BN_B(X)$ = {1,2,3,5}

## Rough Sets - Dependency of Attributes

Let C and D be subsets of A. We say that D depends on C in a degree k ($0 \leq k \leq 1$) denoted by $C \rightarrow_k D$ if

$$k = \gamma(C,\ D) = \sum_{X \in U/D} \frac{|\underline{C}(X)|}{|U|}$$

$$k = \gamma(C,\ D) = \frac{|POS_C(D)|}{|U|}$$

where $POS_C(D) = \bigcup_{X \in U/D} \underline{C}(X)$

If k = 1 we say that D de[pends totally on C

If k , 1 we say that D depends partially (in a degree k) on C



## Rough Sets

| U | Category | Major | Birth_Place | Grade |
|---|---|---|---|---|
| 1 | PhD | History | Detroit | A |
| 2 | MS | Chemistry | Akron | A |
| 3 | MS | History | Detroit | C |
| 4 | BS | Math | Detroit | B |
| 5 | BS | Chemistry | Akron | C |
| 6 | PhD | Computing | Cleveland | A |
| 7 | BS | Chemistry | Cleveland | C |
| 8 | PhD | Computing | Akron | A |

U\{Category} = {{1, 6, 8}, {2, 3}, {4, 5, 7}}
U\{Major} = {{1, 3}, {2, 5, 7}, {4}, {6, 8}}
U\{Birth_Place} = {{2, 5, 8}, {1, 3, 4}, {6, 7}}
U\{Grade} = {{1, 2, 6, 8}, {4}, {3, 5, 7}}

C = {Major, Birth_Place}

U\C = {{1, 3}, {2,5}, {4}.{7}, {8}}

U\D = {{1, 2, 6, 8}, {3, 5, 7}}

D = {Grade}

C({1, 2, 6, 8}) = {6, 8}, $\underline{C}$({4}) =

$\underline{C}$({3, 5, 7}) = {7}

$C \rightarrow_k D = |POS_C(D)|/|U| + 4/8 = 1/2$



## Rough Sets – Dispensable and Indispensable Attributes

Let $S = \{U, A = C \cup D$ be a decision table

Let c be an attribute in C.

Attribute c is dispensable in S if $POS_C(D) = POS_{(C-\{c\})}(D)$

Otherwise, c is indispensable

A decision tale S is independent if all attributes in C are indispensible.



## Rough Sets – Reducts and Core

Let $S = \{U, A = C \cup D$ be a decision table

A subset R of C is a reduct of C, if

$POS_R(D) = POS_C(D)$ and

$S' = (U, R \cup D)$ is independent, i.e., all attributes in R are indispensable in S'.

Core of C is the set of attributes shared by all reducts of C.

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

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



# Rough Sets – Reducts and Core

| U | Category | Major | Birth_Place | Grade |
|---|---|---|---|---|
| 1 | PhD | History | Detroit | A |
| 2 | MS | Chemistry | Akron | A |
| 3 | MS | History | Detroit | C |
| 4 | BS | Math | Detroit | B |
| 5 | BS | Chemistry | Akron | C |
| 6 | PhD | Computing | Cleveland | A |
| 7 | BS | Chemistry | Cleveland | C |
| 8 | PhD | Computing | Akron | A |

U\{Category} = {{1, 6, 8}, {2, 3}, {4, 5, 7}}
U\{Major} = {{1, 3}, {2, 5, 7}, {4}, {6, 8}}
U\{Birth_Place} = {{2, 5, 8}, {1, 3, 4}, {6, 7}}
U\{Grade} = {{1, 2, 6, 8}, {4}, {3, 5, 7}}

## Rough Sets – Rough Membership Function

Let S = (U, A), $B \subseteq A$ and $X \subseteq U$

Then the rough membership function $\mu B_X$ for X is a mapping from U to [0, 1]

$$\mu B_X : U \rightarrow [0, 1]$$

For all e in U, the degree of e belongs to X in light of the set of attributes B is defined as

$$\mu B_X (e) = |B(e) \cap X|/B(e)$$

where B(e) denotes the block containing e.

## Concluding Remarks

### Out of time

My old clock used to tell the time

and subdivide diurnity;

but now it's lost both hands and chime

and only tells eterni