„Aurel Vlaicu” University of Arad
Faculty of Engineering

Scientific and Technical Bulletin

Series: Electrotechnics, Electronics, Automatic Control and Computer Science

Vol. 2, No. 1, 2005

ISSN 1584-9198

Honorary Editor:
Lotfi A. Zadeh – University of California, Berkeley (SUA)

Editor-in-Chief
Valentina E. Balaş – „Aurel Vlaicu” University of Arad (Romania)

Editorial Board:
Jair Minoro Abe – Instituto de Ciências Exatas e Tecnologia, São Paulo (Brazil)
Marius M. Balaş – „Aurel Vlaicu” University of Arad (Romania)
Nicolae Budişan – Politehnica University of Timişoara (Romania)
Chihab Hanachi – IRIT Laboratory, University Toulouse (France)
Cornel Barna – „Aurel Vlaicu” University of Arad (Romania)
Mircea Ciugudean – Politehnica University of Timişoara (Romania)
Mihaela Costin – Romanian Academy, Computer Science Institute, Iaşi (Romania)
Sheng-Luen Chung – NTUST Taipei (Taiwan)
Radu Dogaru – Politehnica University of Bucharest (Romania)
Toma L. Dragomir – Politehnica University of Timişoara (Romania)
Jean Duplaix – Université du Sud Toulon-Var, Toulon (France)
Michael C. Fairhurst – University of Kent (UK)
Florin Filip – Romanian Academy (Romania)
Janos Fodor – Budapest Tech (Hungary)
Voicu Groza – University of Ottawa (Canada)
Hacene Habbi – University of Boumerdès (Algier)
Jan Jantzen – Technical University of Denmark, Kongens Lyngby (Denmark)
Lakhmi C. Jain – University of South Australia Adelaide (Australia)

Laszlo T. Koczy – University of T. E., Budapest and S. I. University Győr (Hungary)
Chung-Hsien Kuo – NTUST Taipei (Taiwan)
Veljko Milutinovic – University of Belgrade (Serbia)
Costin Miron – Technical University Cluj-Napoca (Romania)
Dorothy N. Monekosso – Kingston University, Kingston upon Thames (UK)
Valentin Muller – „Aurel Vlaicu" University of Arad (Romania)
Kazumi Nakamatsu – University of Hyogo (Japan)
Viorel Nicolau – Dunărea de Jos University of Galați (Romania)
Onisifor Olaru – „Constantin Brancussy" University, Tg.-Jiu (Romania)
Stephan Olariu – Old Dominion University, Norfolk (U. S. A.)
Emil Petriu – University of Ottawa (Canada)
Octavian Proştean – Politehnica University of Timişoara (Romania)
Felix Riesco-Pelaez – University of León (Spain)
Daniela E. Popescu – University of Oradea (Romania)
Dumitru Popescu – Politehnica University of Bucharest (Romania)
Ales Prochazka – Institute of Chemical Technology Prague (Czech Republic)
Sahondranirina Ravonialimanana – Universite de Fianarantsoa (Madagascar)
Anca Ralescu – University of Cincinnati (U. S. A.)
Dan Ralescu – University of Cincinnati (U. S. A.)
Imre Rudas – Budapest Tech (Hungary)
Rudolf Seising – European Centre for Soft Computing, Mieres, (Spain)
Kostas Sirlantzis – University of Kent (UK)
Tiberiu Spircu – „Carol Davila" University of Medicine and Pharmacy, Bucarest (Romania)
Michio Sugeno – Doshisha University, Kyoto (Japan)
Horia Nicolai Teodorescu – „Gheorghe Asachi" Technical University Iaşi (Romania)

Mohamed Tounsi – Prince Sultan University, Riyadh (Saudi Arabia)
Annamaria Varkoniy-Koczy – BUTE Budapest (Hungary)
Mircea Vlăduțiu – Politehnica University of Timişoara (Romania)
Djuro G. Zrilic – New Mexico Highlands University (U. S. A.)

Editorial Manager

Marius Buzera – Colegiul Tehnic „Gheorghe Magheru”, Tg.-Jiu (Romania)

Aims & scope

The *Electrotechnics, Electronics, Automatic Control and Computer Science* series of the Scientifical and Technical Bulletin of the „Aurel Vlaicu” University of Arad will devote it self to the dissemination of the original research results, technical advances and new items in Electrical and Computers Engineering and in Knowledge Engineering.

The team of the *Automate Control and Applied Software Department* of the above denominated academic institution is intending to build mutual benefic interactions with researchers and institutions in the field.

Published 4 times a year

All papers are refereed through a double blind process. A guide for authors, sample copies and other relevant information for submitting papers are available at **http://uavsb.xhost.ro**

Please send the submitted paper via e-mail to:
Dr. Valentina E. Balas
http://www.drbalas.ro/uav_scientific_bulletin.htm

ISSN 1584-9198

Table of Contents

Lotfi A. ZADEH

Professor in the Graduate School
Director, Berkeley Initiative in Soft Computing (BISC)
Address: 729 Soda Hall #1776
Computer Science Division
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720-1776
E-mail: *zadeh@eecs.berkeley.edu*

COMPUTING WITH WORDS AND PERCEPTIONS (CWP) – A SHIFT IN DIRECTION IN COMPUTING AND DECISION ANALYSIS

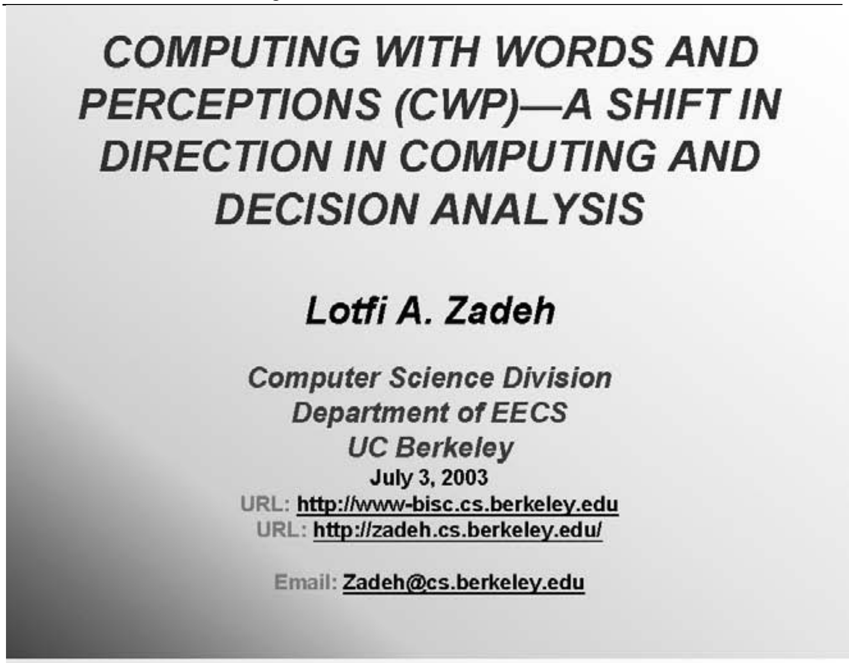
Note: *The paper was presented during the ceremony of conferring honorary degree to Prof. Lotfi A. Zadeh at „Aurel Vlaicu" University of Arad, Romania, Arad, 03.07.2003*

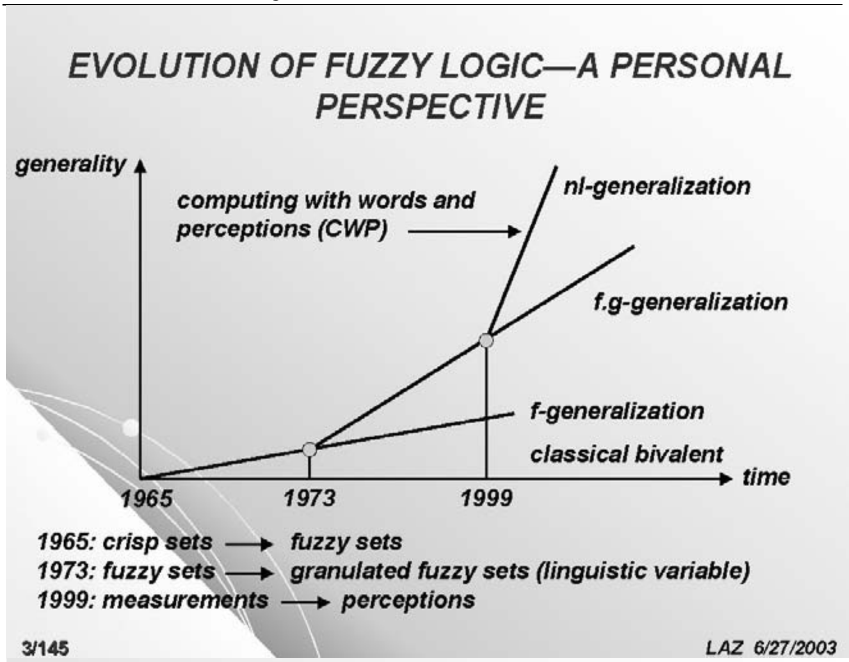
Abstract

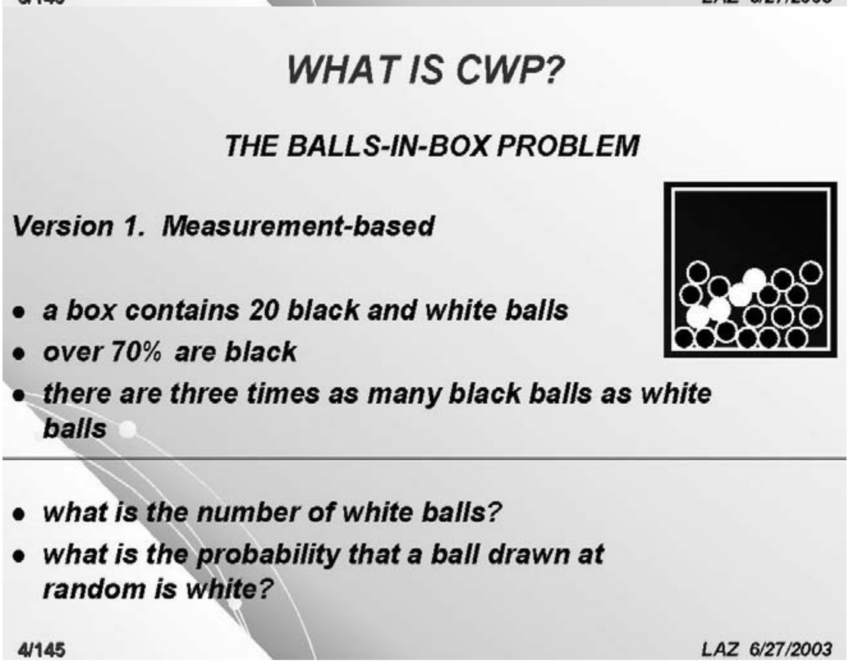
In computing with words and perceptions, or CWP for short, the objects of computation are words, propositions and perceptions described in a natural language. In science, there is a deep-seated tradition of striving for progression from perceptions to measurements, and from the use of words to the use of numbers.
Reflecting the bounded ability of sensory organs and, ultimately, the brain, to resolve detail, perceptions are intrinsically imprecise. Perceptions are f-granular in the sense that (a) the perceived values of attributes are fuzzy; and (b) the perceived values of attributes are granular, with a granule being a clump of values drawn together by indistinguishability, similarity, proximity or functionality.
F-granularity of perceptions is the reason why in the enormous literature on perceptions one cannot find a theory in which perceptions are objects of computation, as they are in CWP.
PNL (precisiated natural language) associates with a natural language, NL, a precisiation language, GCL (Generalized Constraint Language), which consists of generalized constraints and their combinations and qualifications.
The principal function of PNL is to serve as a system for computation and reasoning with perceptions. The need for redefinition arises because standard bivalent – classic-based definitions may lead to counterintuitive conclusions.
Computing with words and perceptions provides a basis for an important generalization of probability theory, namely, perception-based probability theory (PTp).
The importance of CWP derives from the fact that it opens the door to adding to any measurement-based theory.

Keywords: *fuzzy, CWP (computing with words), PNL (precisiated natural language), probability theory.*

Scientific and Technical Bulletin Series: Electrotechnics, Electronics, Automatic Control and Computer Science, Vol. 2, No. 1, 2005, ISSN 1584-9198




EVOLUTION OF FUZZY LOGIC—A PERSONAL PERSPECTIVE
generality
computing with words and perceptions (CWP)
nl-generalization
f.g-generalization
f-generalization
classical bivalent
time
1965
1973
1999
1965: crisp sets → fuzzy sets
1973: fuzzy sets → granulated fuzzy sets (linguistic variable)
1999: measurements → perceptions
3/145
LAZ 6/27/2003


WHAT IS CWP?
THE BALLS-IN-BOX PROBLEM
Version 1. Measurement-based
• a box contains 20 black and white balls
• over 70% are black
• there are three times as many black balls as white balls
• what is the number of white balls?
• what is the probability that a ball drawn at random is white?
4/145
LAZ 6/27/2003

CONTINUED

Version 2. Perception-based

- *a box contains about 20 black and white balls*
- *most are black*
- *there are several times as many black balls as white balls*

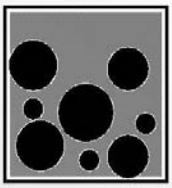
- *what is the number of white balls?*
- *what is the probability that a ball drawn at random is white?*

5/145 LAZ 6/27/2003

CONTINUED

Version 3. Perception-based

box



- *a box contains about 20 black balls of various sizes*
- *most are large*
- *there are several times as many large balls as small balls*

- *what is the number of small balls?*
- *what is the probability that a ball drawn at random is small?*

6/145 LAZ 6/27/2003

MEASUREMENT-BASED

- *a box contains 20 black and white balls*
- *over seventy percent are black*
- *there are three times as many black balls as white balls*
- *what is the number of white balls?*
- *what is the probability that a ball picked at random is white?*

PERCEPTION-BASED (version 1)

- *a box contains about 20 black and white balls*
- *most are black*
- *there are several times as many black balls as white balls*
- *what is the number of white balls*
- *what is the probability that a ball drawn at random is white?*

7/145 LAZ 6/27/2003

COMPUTATION (version 1)

- *measurement-based*

  X = number of black balls

  Y_2 number of white balls

  $X \geq 0.7 \cdot 20 = 14$

  $X + Y = 20$

  $X = 3Y$

  $X = 15$; $Y = 5$

  $p = 5/20 = .25$

- *perception-based*

  X = number of black balls

  Y = number of white balls

  $X = most \times 20^*$

  $X = several \ {}^*Y$

  $X + Y = 20^*$

  $P = Y/N$

8/145 LAZ 6/27/2003

BASIC PERCEPTIONS

attributes of physical objects

•*distance*	•*length*	•*weight*
•*time*	•*width*	•*height*
•*speed*	•*area*	•*size*
•*direction*	•*volume*	•*temperature*

sensations and emotions

•*color*	•*hunger*	•*joy*
•*smell*	•*thirst*	•*anger*
•*pain*	•*cold*	•*fear*

concepts

•*count*	•*causality*	•*truth*
•*similarity*	•*relevance*	•*likelihood*
•*cluster*	•*risk*	•*possibility*

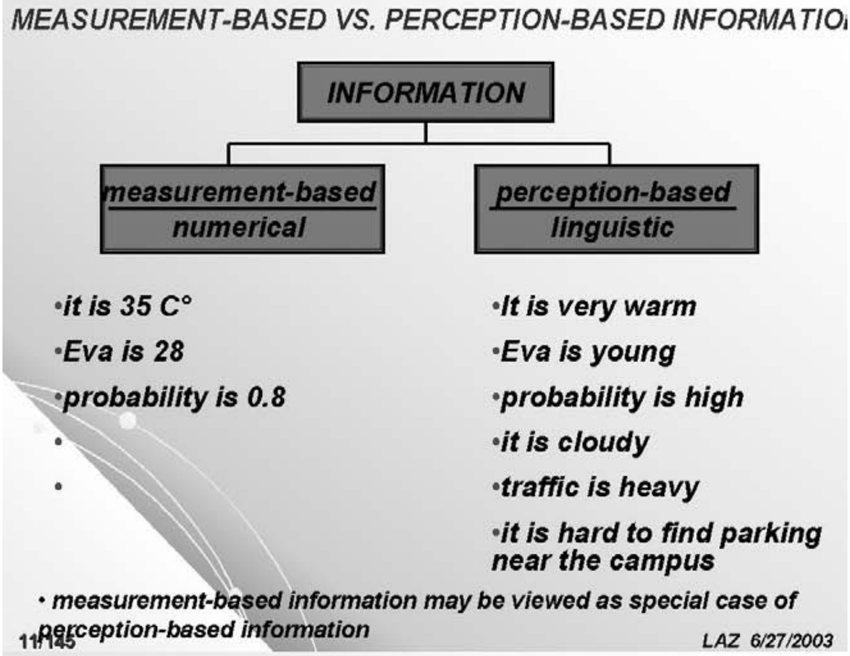
9/145 LAZ 6/27/2003

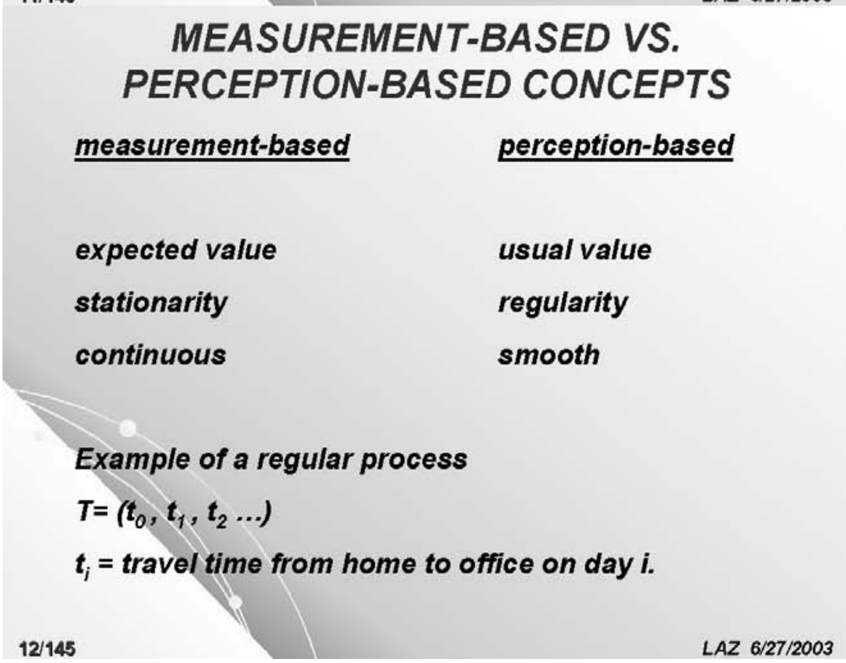
DEEP STRUCTURE OF PERCEPTIONS

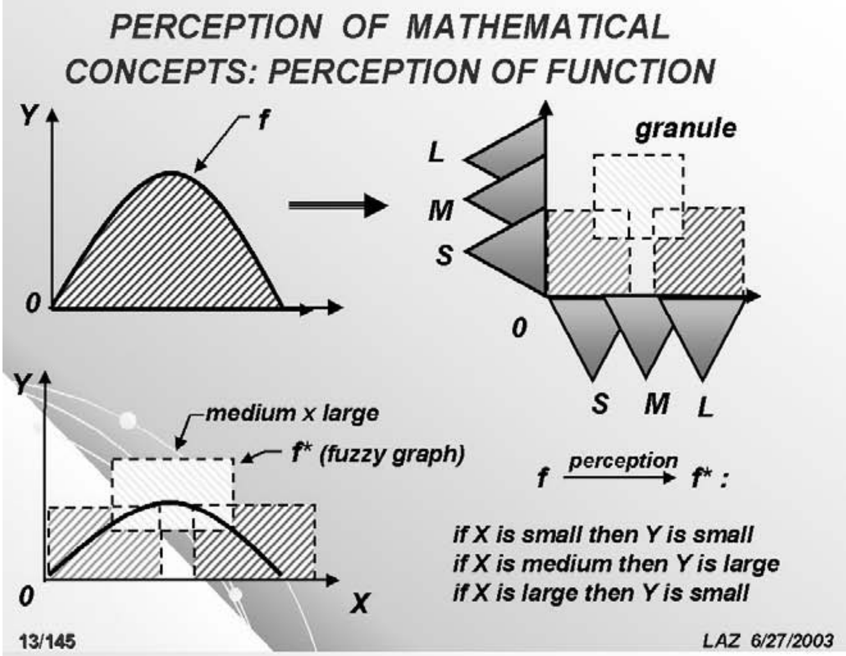
- *perception of likelihood*
- *perception of truth (compatibility)*
- *perception of possibility (ease of attainment or realization)*
- *perception of similarity*
- *perception of count (absolute or relative)*
- *perception of causality*

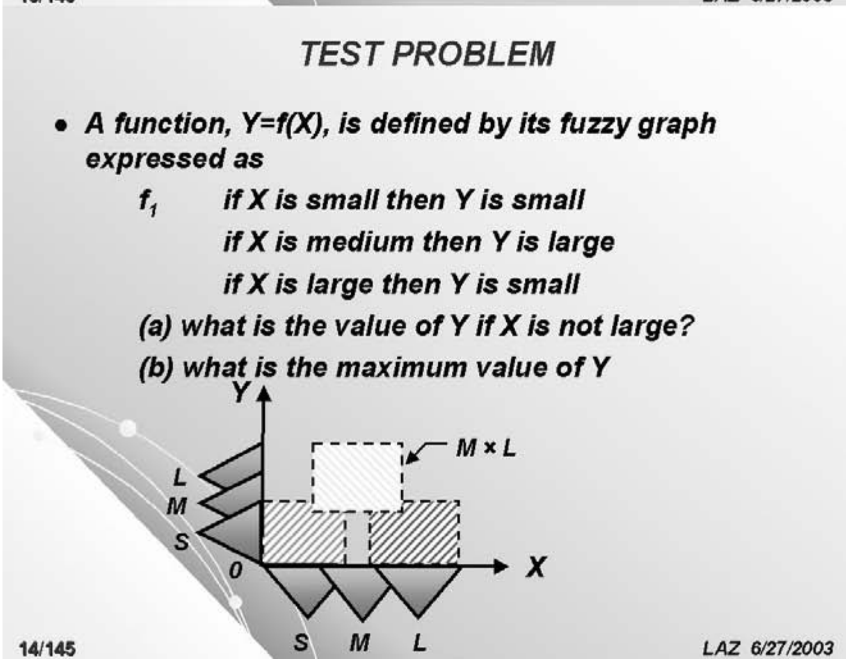
subjective probability = quantification of perception of likelihood

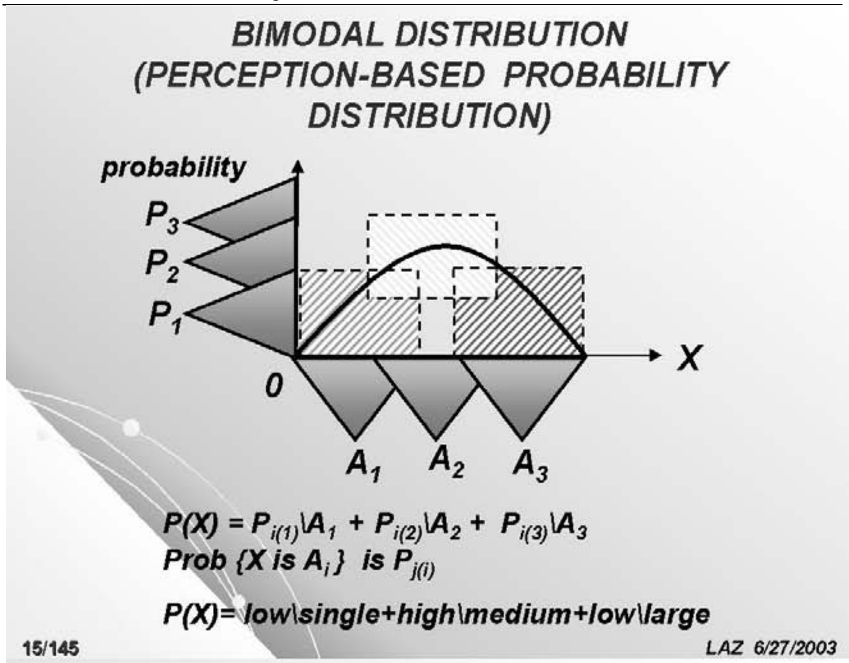
10/145 LAZ 6/27/2003

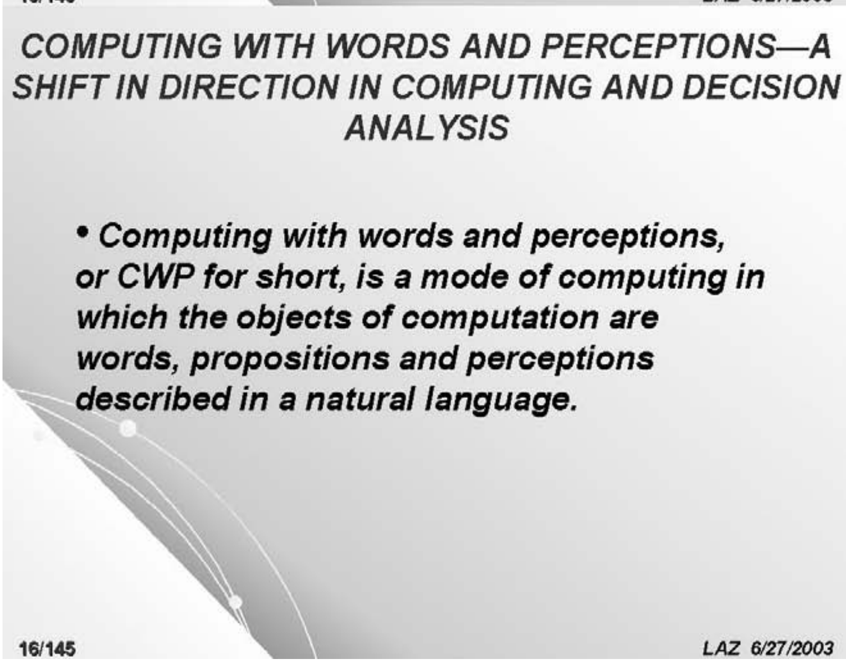

MEASUREMENT-BASED VS. PERCEPTION-BASED INFORMATIO
INFORMATION
measurement-based
numerical
perception-based
linguistic
•it is 35 C°
•Eva is 28
•probability is 0.8
•It is very warm
•Eva is young
•probability is high
•it is cloudy
•traffic is heavy
•it is hard to find parking near the campus
• measurement-based information may be viewed as special case of perception-based information
11/145
LAZ 6/27/2003


MEASUREMENT-BASED VS. PERCEPTION-BASED CONCEPTS
measurement-based
perception-based
expected value
usual value
stationarity
regularity
continuous
smooth
Example of a regular process
$T = (t_0, t_1, t_2 \ldots)$
t_i = travel time from home to office on day i.
12/145
LAZ 6/27/2003


PERCEPTION OF MATHEMATICAL CONCEPTS: PERCEPTION OF FUNCTION
Y
f
0
granule
L
M
S
0
S M L
Y
medium x large
f* (fuzzy graph)
0
X
f perception f* :
if X is small then Y is small
if X is medium then Y is large
if X is large then Y is small
13/145
LAZ 6/27/2003


TEST PROBLEM
• A function, Y=f(X), is defined by its fuzzy graph expressed as
f1 if X is small then Y is small
if X is medium then Y is large
if X is large then Y is small
(a) what is the value of Y if X is not large?
(b) what is the maximum value of Y
Y
M × L
L
M
S
0
X
S M L
14/145
LAZ 6/27/2003


BIMODAL DISTRIBUTION
(PERCEPTION-BASED PROBABILITY
DISTRIBUTION)
probability
P3
P2
P1
0
X
A1
A2
A3
P(X) = Pi(1)\A1 + Pi(2)\A2 + Pi(3)\A3
Prob {X is Ai} is Pj(i)
P(X)= low\single+high\medium+low\large
15/145
LAZ 6/27/2003


COMPUTING WITH WORDS AND PERCEPTIONS—A SHIFT IN DIRECTION IN COMPUTING AND DECISION ANALYSIS
• Computing with words and perceptions, or CWP for short, is a mode of computing in which the objects of computation are words, propositions and perceptions described in a natural language.
16/145
LAZ 6/27/2003

CONTINUED

- ***Perceptions play a key role in human cognition. Humans—but not machines—have a remarkable capability to perform a wide variety of physical and mental tasks without any measurements and any computations. Everyday examples of such tasks are driving a car in city traffic, playing tennis and summarizing a book.***

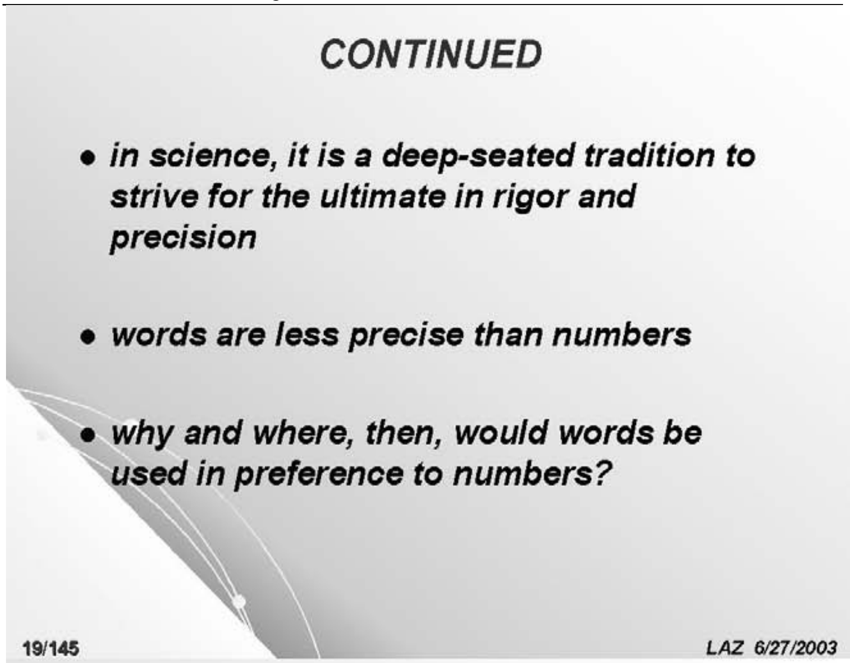
17/145 LAZ 6/27/2003

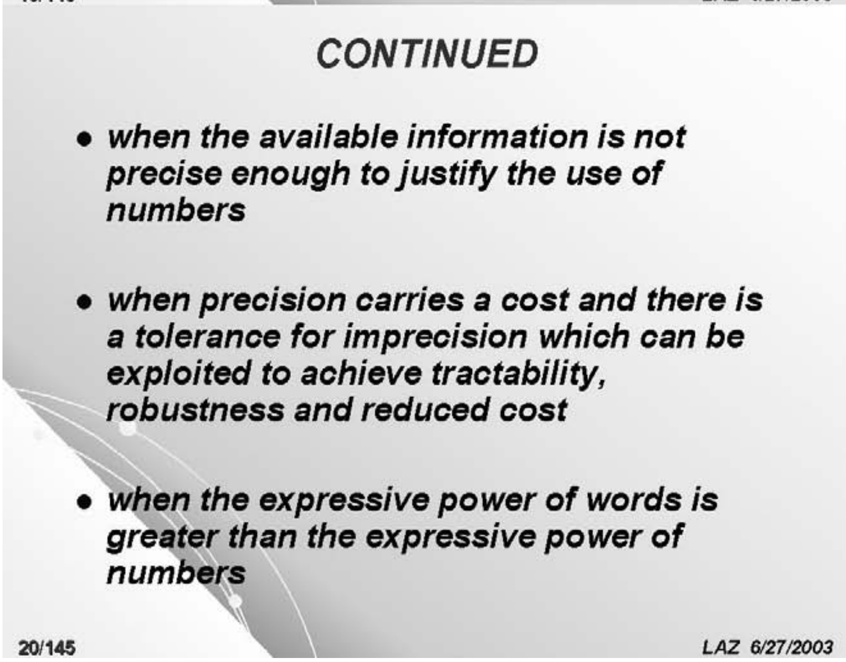
COMPUTING WITH WORDS AND PERCEPTIONS (CWP)

Key points

- ***In computing with words and perceptions, the objects of computation are words, propositions, and perceptions described in a natural language***
- ***A natural language is a system for describing perceptions***
- ***In CWP, a perception is equated to its description in a natural language***

18/145 LAZ 6/27/2003


CONTINUED
• in science, it is a deep-seated tradition to strive for the ultimate in rigor and precision
• words are less precise than numbers
• why and where, then, would words be used in preference to numbers?
19/145
LAZ 6/27/2003


CONTINUED
• when the available information is not precise enough to justify the use of numbers
• when precision carries a cost and there is a tolerance for imprecision which can be exploited to achieve tractability, robustness and reduced cost
• when the expressive power of words is greater than the expressive power of numbers
20/145
LAZ 6/27/2003

CONTINUED

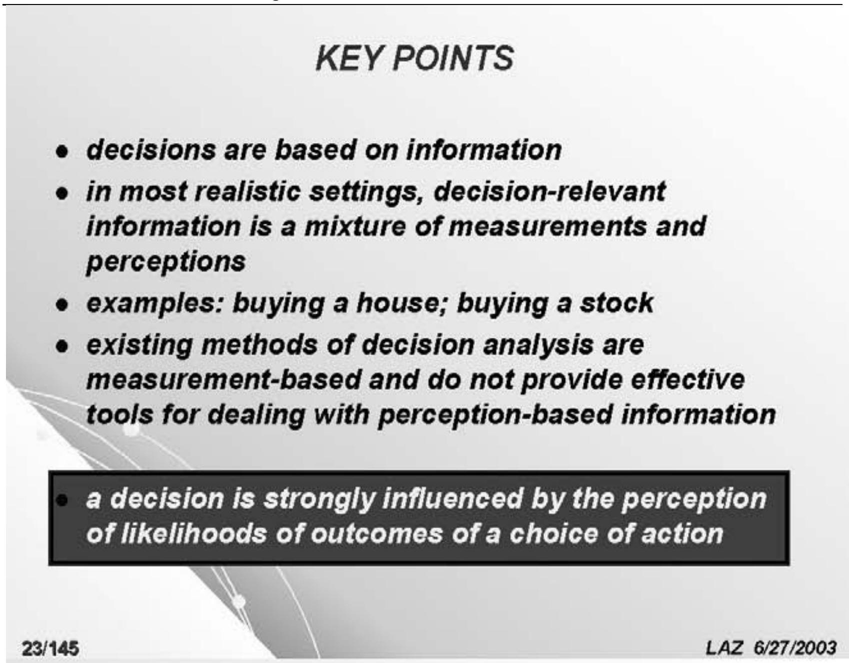
- *One of the major aims of CWP is to serve as a basis for equipping machines with a capability to operate on perception-based information. A key idea in CWP is that of dealing with perceptions through their descriptions in a natural language. In this way, computing and reasoning with perceptions is reduced to operating on propositions drawn from a natural language.*

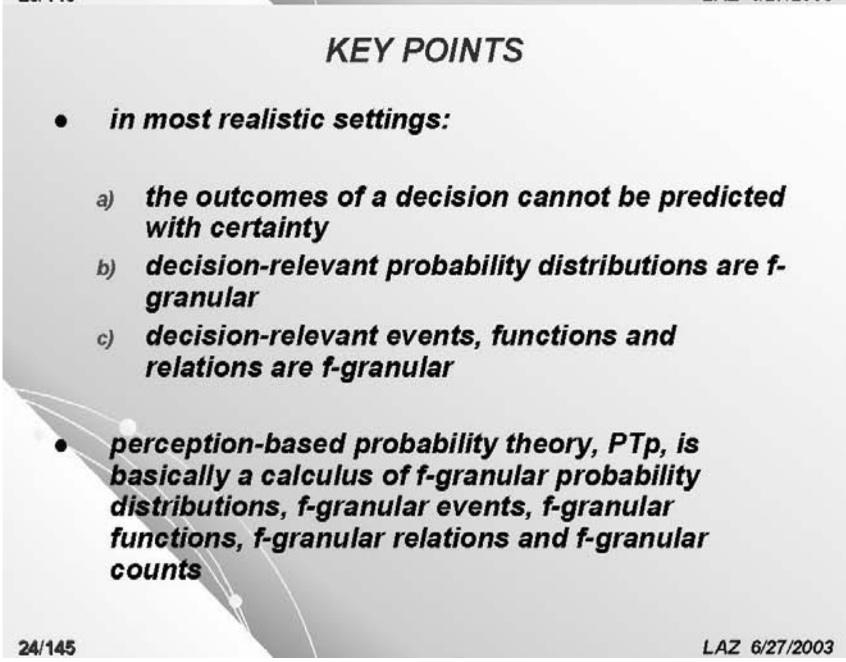
21/145 LAZ 6/27/2003

CONTINUED

- *In CWP, what is employed for this purpose is PNL (Precisiated Natural Language.) In PNL, a proposition, p, drawn from a natural language, NL, is represented as a generalized constraint, with the language of generalized constraints, GCL, serving as a precisiation language for computation and reasoning, PNL is equipped with two dictionaries and a modular multiagent deduction database. The rules of deduction are expressed in what is referred to as the Protoform Language (PFL).*

22/145 LAZ 6/27/2003


KEY POINTS
• decisions are based on information
• in most realistic settings, decision-relevant information is a mixture of measurements and perceptions
• examples: buying a house; buying a stock
• existing methods of decision analysis are measurement-based and do not provide effective tools for dealing with perception-based information
• a decision is strongly influenced by the perception of likelihoods of outcomes of a choice of action
23/145
LAZ 6/27/2003


KEY POINTS
• in most realistic settings:
a) the outcomes of a decision cannot be predicted with certainty
b) decision-relevant probability distributions are f-granular
c) decision-relevant events, functions and relations are f-granular
• perception-based probability theory, PTp, is basically a calculus of f-granular probability distributions, f-granular events, f-granular functions, f-granular relations and f-granular counts
24/145
LAZ 6/27/2003

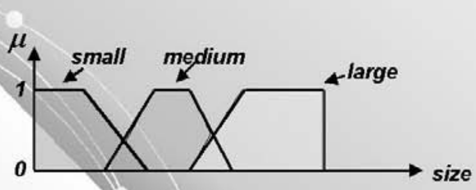
OBSERVATION

- *machines are driven by measurements*
- *humans are driven by perceptions*
- *to enable a machine to mimic the remarkable human capability to perform a wide variety of physical and mental tasks using perception-based information, it is necessary to have a means of converting measurements into perceptions*

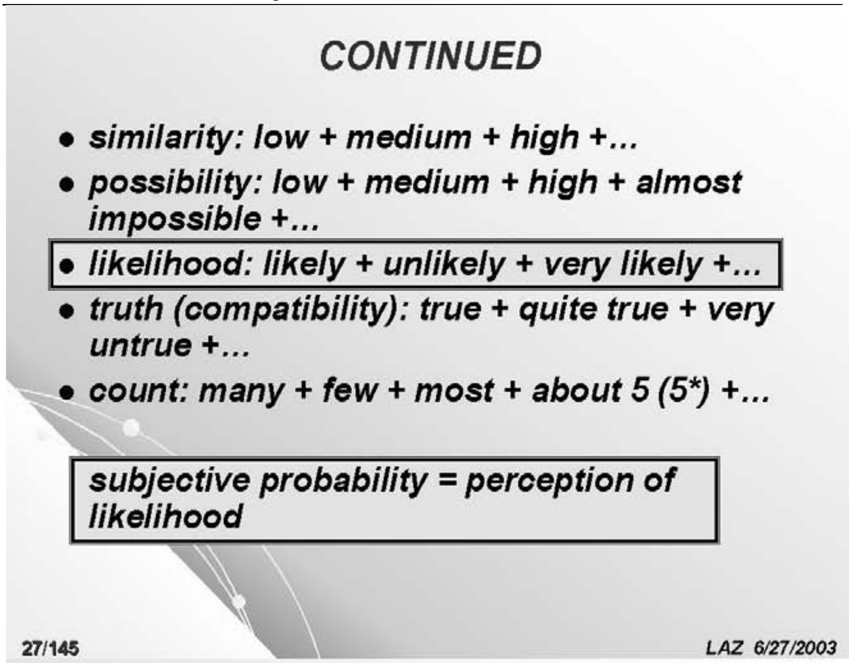
25/145 LAZ 6/27/2003

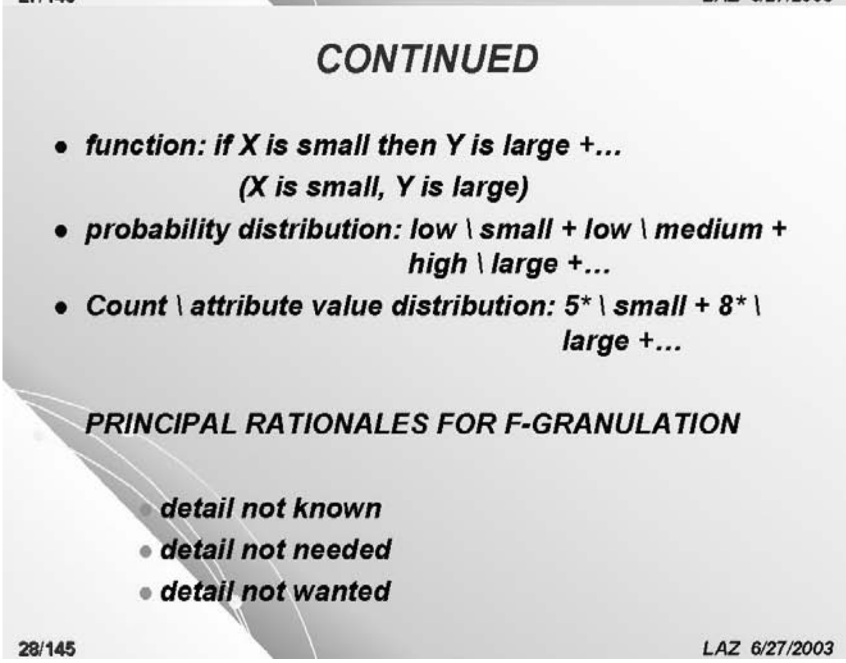
BASIC PERCEPTIONS / F-GRANULARITY

- *temperature: warm+cold+very warm+much warmer+...*
- *time: soon + about one hour + not much later +...*
- *distance: near + far + much farther +...*
- *speed: fast + slow +much faster +...*
- *length: long + short + very long +...*

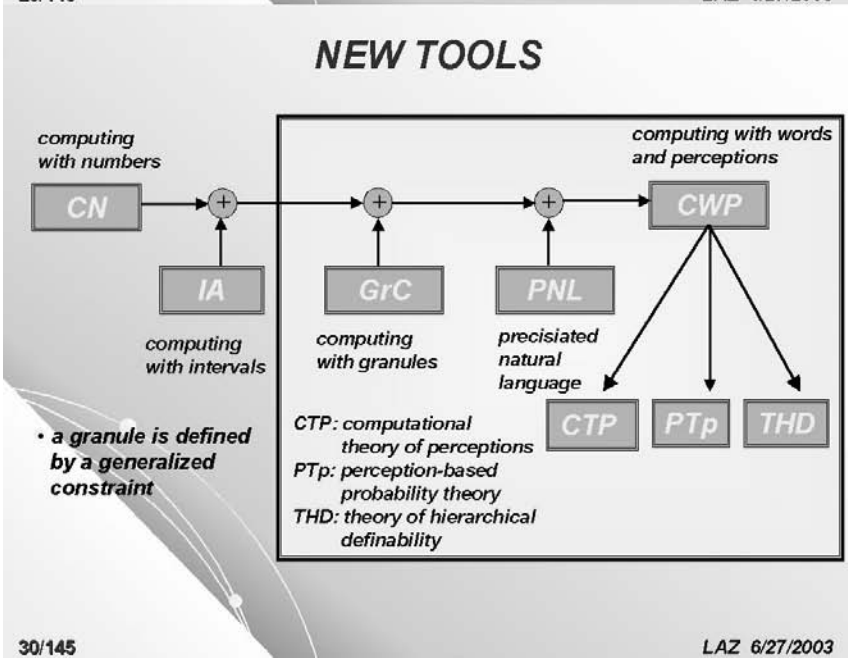


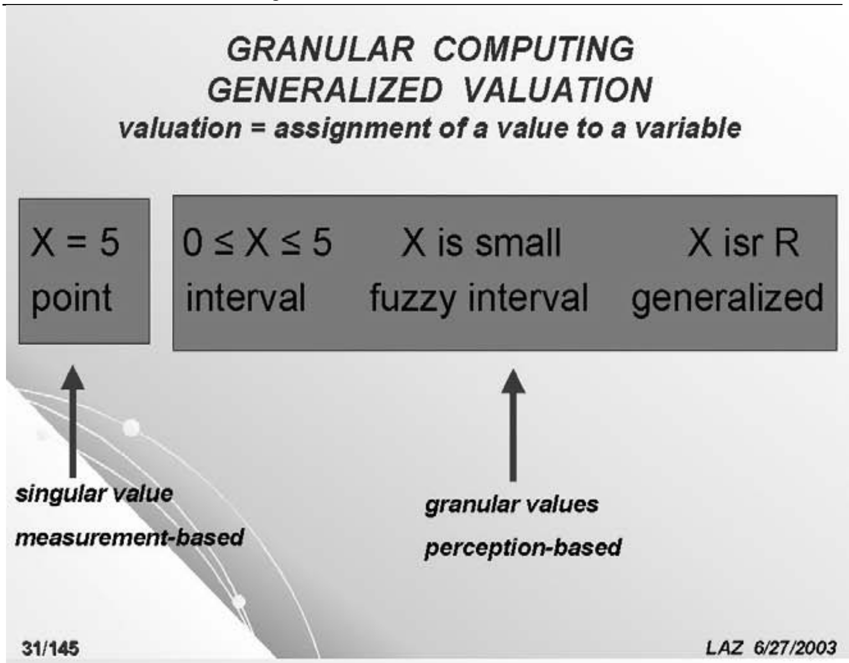
26/145 LAZ 6/27/2003

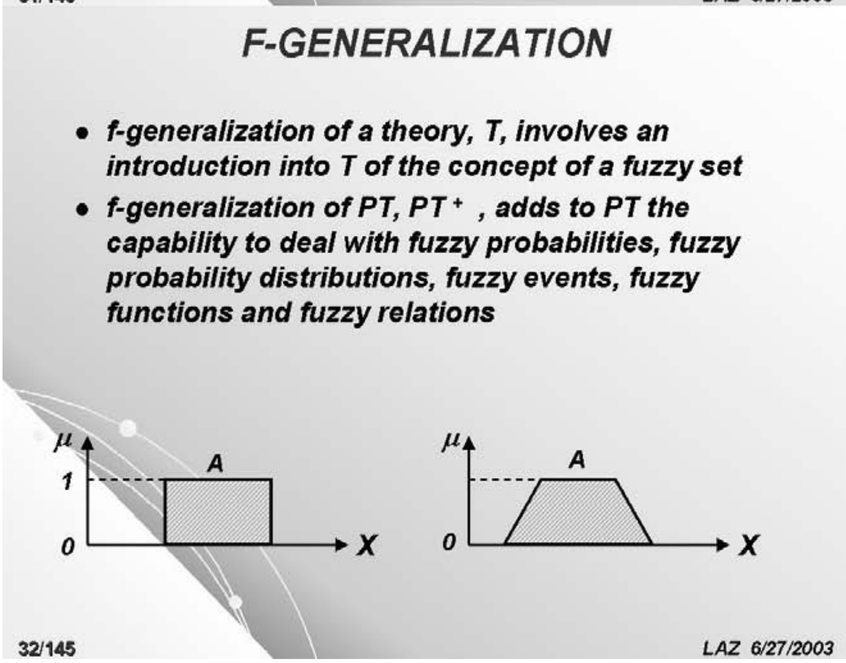

CONTINUED
• similarity: low + medium + high +...
• possibility: low + medium + high + almost impossible +...
• likelihood: likely + unlikely + very likely +...
• truth (compatibility): true + quite true + very untrue +...
• count: many + few + most + about 5 (5*) +...
subjective probability = perception of likelihood
27/145
LAZ 6/27/2003


CONTINUED
• function: if X is small then Y is large +...
(X is small, Y is large)
• probability distribution: low \ small + low \ medium + high \ large +...
• Count \ attribute value distribution: 5* \ small + 8* \ large +...
PRINCIPAL RATIONALES FOR F-GRANULATION
• detail not known
• detail not needed
• detail not wanted
28/145
LAZ 6/27/2003


New Tools
29/145
LAZ 6/27/2003

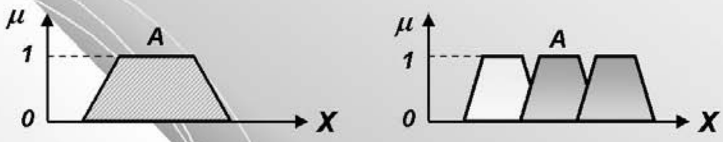

NEW TOOLS
computing
with numbers
CN
IA
computing
with intervals
GrC
computing
with granules
PNL
precisiated
natural
language
computing with words
and perceptions
CWP
CTP
PTp
THD
• a granule is defined
by a generalized
constraint
CTP: computational
theory of perceptions
PTp: perception-based
probability theory
THD: theory of hierarchical
definability
30/145
LAZ 6/27/2003


GRANULAR COMPUTING
GENERALIZED VALUATION
valuation = assignment of a value to a variable
X = 5
point
0 ≤ X ≤ 5
interval
X is small
fuzzy interval
X isr R
generalized
singular value
measurement-based
granular values
perception-based
31/145
LAZ 6/27/2003


F-GENERALIZATION
• f-generalization of a theory, T, involves an introduction into T of the concept of a fuzzy set
• f-generalization of PT, PT +, adds to PT the capability to deal with fuzzy probabilities, fuzzy probability distributions, fuzzy events, fuzzy functions and fuzzy relations
μ
1
0
A
X
μ
0
A
X
32/145
LAZ 6/27/2003

F.G-GENERALIZATION

- ***f.g-generalization of T, T^{++}, involves an introduction into T of the concept of a granulated fuzzy set***
- ***f.g-generalization of PT, PT^{++} , adds to PT^{+} the capability to deal with f-granular probabilities, f-granular probability distributions, f-granular events, f-granular functions and f-granular relations***



33/145 LAZ 6/27/2003

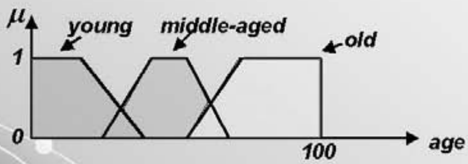
EXAMPLES OF F-GRANULATION (LINGUISTIC VARIABLES)

color: red, blue, green, yellow, …

age: young, middle-aged, old, very old

size: small, big, very big, …

distance: near, far, very, not very far, …



- ***humans have a remarkable capability to perform a wide variety of physical and mental tasks, e.g., driving a car in city traffic, without any measurements and any computations***
- ***one of the principal aims of CTP is to develop a better understanding of how this capability can be added to machines***

34/145 LAZ 6/27/2003

PRECISIATED NATURAL LANGUAGE

PNL

36/145 LAZ 6/27/2003

WHAT IS PRECISIATED NATURAL LANGUAGE (PNL)? PRELIMINARIES

- a proposition, p, in a natural language, NL, is precisiable if it translatable into a precisiation language
- in the case of PNL, the precisiation language is the Generalized Constraint Language, GCL
- precisiation of p, p*, is an element of GCL (GC-form)

37/145 LAZ 6/27/2003

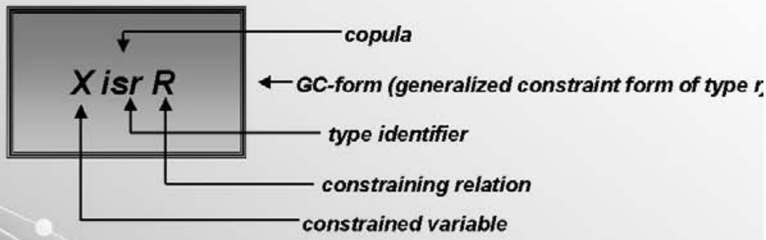
WHAT IS PNL?

- ***PNL is a sublanguage of precisiable propositions in NL which is equipped with two dictionaries: (1) NL to GCL; (2) GCL to PFL (Protoform Language); and (3) a modular multiagent database of rules of deduction (rules of generalized constrained propagation) expressed in PFL.***

38/145 *LAZ 6/27/2003*

GENERALIZED CONSTRAINT

•***standard constraint:*** **X** $\in$ **C**
•***generalized constraint: X isr R***



•***X= (X₁ , …, Xₙ)***
•***X may have a structure: X=Location (Residence(Carol))***
•***X may be a function of another variable: X=f(Y)***
•***X may be conditioned: (X/Y)***
• **r := / ≤ / ... / ⊂ / ⊃ / blank / v / p / u / rs / fg / ps / ...**

39/145 *LAZ 6/27/2003*

GC-FORM (GENERALIZED CONSTRAINT FORM OF TYPE r)

X isr R

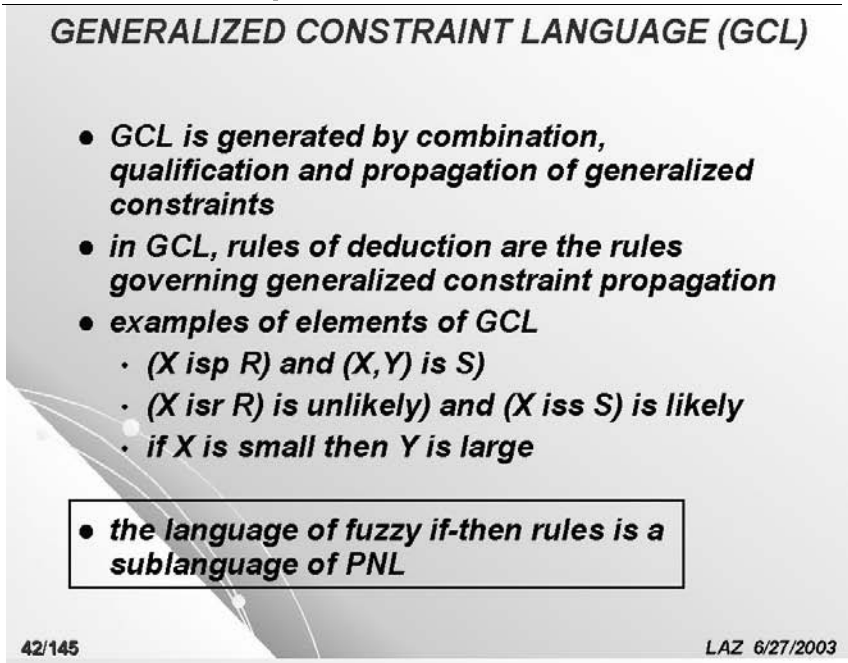
r: =	***equality constraint: X=R is abbreviation of X is=R***
r: ≤	***inequality constraint: X ≤ R***
r:⊂	***subsethood constraint: X ⊂ R***
r: blank	***possibilistic constraint; X is R; R is the possibility distribution of X***
r: v	***veristic constraint; X isv R; R is the verity distribution of X***
r: p	***probabilistic constraint; X isp R; R is the probability distribution of X***

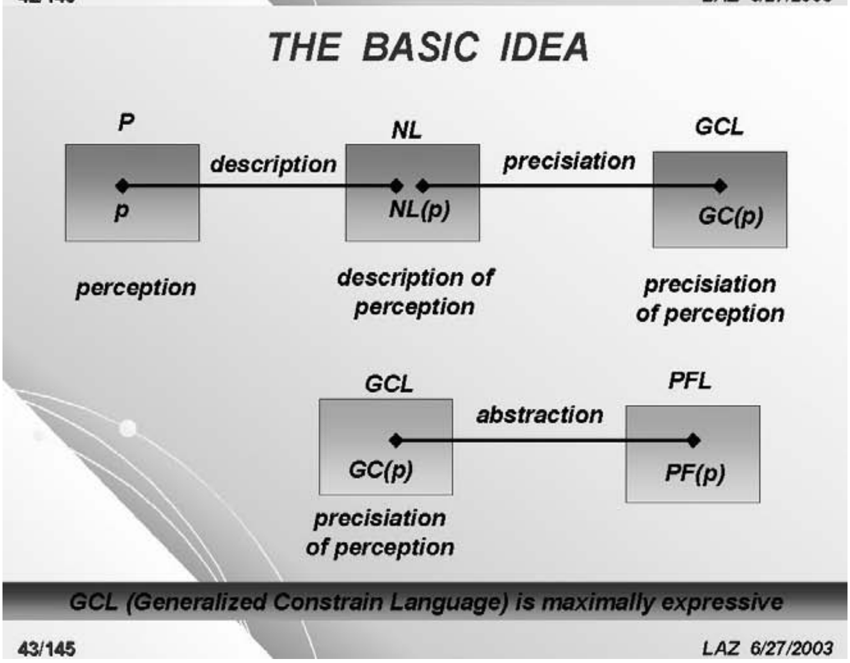
40/145 LAZ 6/27/2003

CONTINUED

r: rs	*random set constraint; X isrs R; R is the set-valued probability distribution of X*
r: fg	*fuzzy graph constraint; X isfg R; X is a function and R is its fuzzy graph*
r: u	*usuality constraint; X isu R means usually (X is R)*
r: ps	*Pawlak set constraint: X isps ($\underline{X}$, $\overline{X}$) means that X is a set and $\underline{X}$ and $\overline{X}$ are the lower and upper approximations to X*

41/145 LAZ 6/27/2003


GENERALIZED CONSTRAINT LANGUAGE (GCL)
GCL is generated by combination, qualification and propagation of generalized constraints
in GCL, rules of deduction are the rules governing generalized constraint propagation
examples of elements of GCL
(X isp R) and (X,Y) is S)
(X isr R) is unlikely) and (X iss S) is likely
if X is small then Y is large
the language of fuzzy if-then rules is a sublanguage of PNL
42/145
LAZ 6/27/2003


THE BASIC IDEA
P
NL
GCL
description
precisiation
p
NL(p)
GC(p)
perception
description of perception
precisiation of perception
GCL
PFL
abstraction
GC(p)
PF(p)
precisiation of perception
GCL (Generalized Constrain Language) is maximally expressive
43/145
LAZ 6/27/2003

WHAT IS A PROTOFORM?

- *Informally, a protoform (abbreviation of "prototypical form") is a symbolic expression which places in evidence the deep semantic structure of a proposition, question or command.*

examples:

X is A $\xrightarrow{\text{instantiation}}$ *Speed is 150 km/h*

X isp A $\xrightarrow{\text{instantiation}}$ *X is a normally distributed random variable with mean m and variance* σ^2

A(B) is C $\xrightarrow{\text{instantiation}}$ *Age(Eva) is young (Eva is young)*

Prob (X is A) is B $\xrightarrow{\text{instantiation}}$ *Usually Robert returns from work at about 6 pm*

A (B, C) is D $\xrightarrow{\text{instantiation}}$ *Distance between Los Angeles and San Francisco is about 600 km.*

44/145 LAZ 6/27/2003

CONTINUED

- *A protoform of p defines its deep semantic structure*

Examples:

- *Allan is tall* → *A(B) is R*
- *distance between New York and Boston is 200 miles* → *A(B, C) is R*
- *Most Swedes are tall* → *Count (A/B) is Q*
- *Usually Robert returns from work at about 6 pm* → *Prob (X is A) is B*

45/145 LAZ 6/27/2003

DICTIONARIES

1:

proposition in NL	precisiation
p	*p** (GC-form)
most Swedes are tall	Σ Count (tall.Swedes/Swedes) is most

2:

precisiation	protoform
*p** (GC-form)	PF(p*)
Σ Count (tall.Swedes/Swedes) is most	Q A's are B's

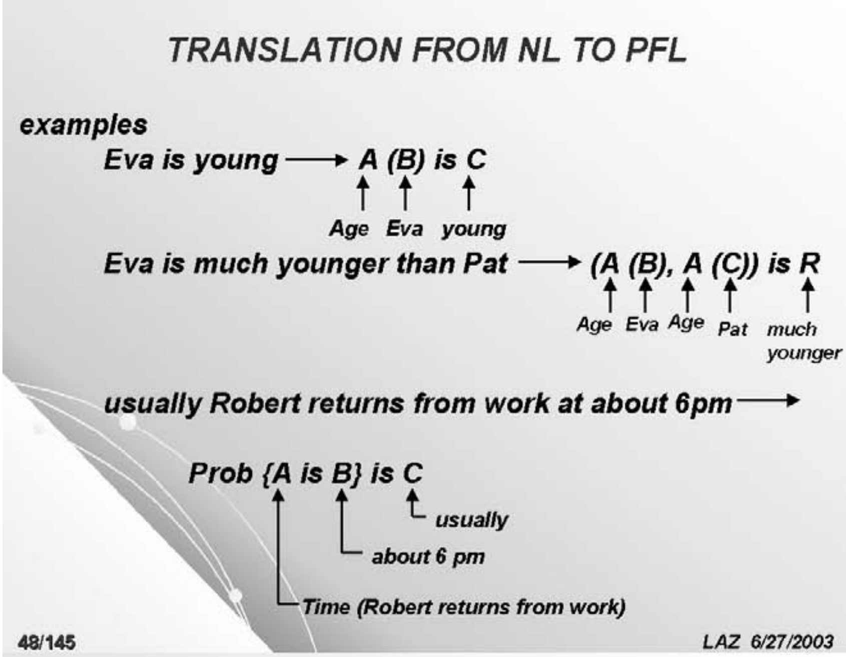
46/145 LAZ 6/27/2003

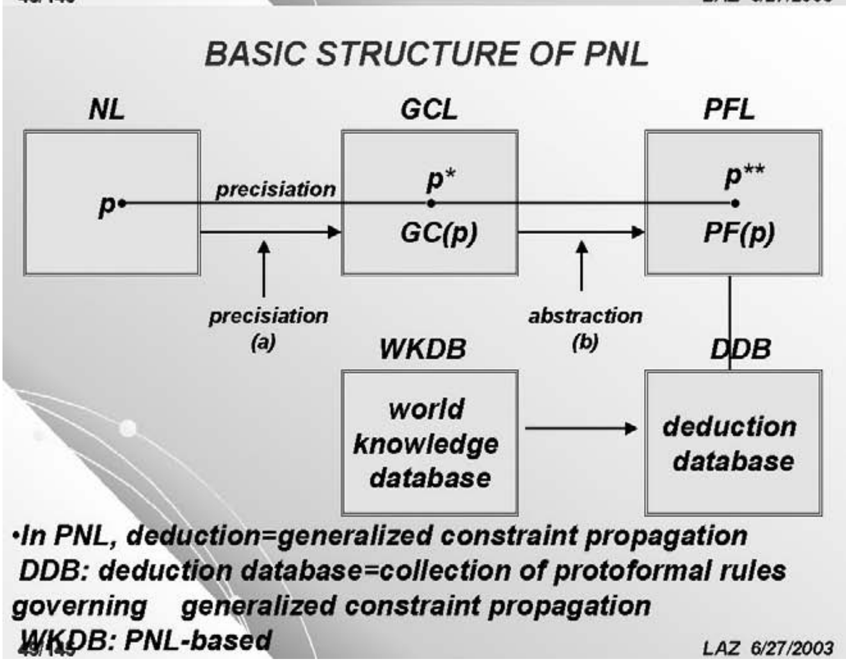
EXAMPLE OF TRANSLATION

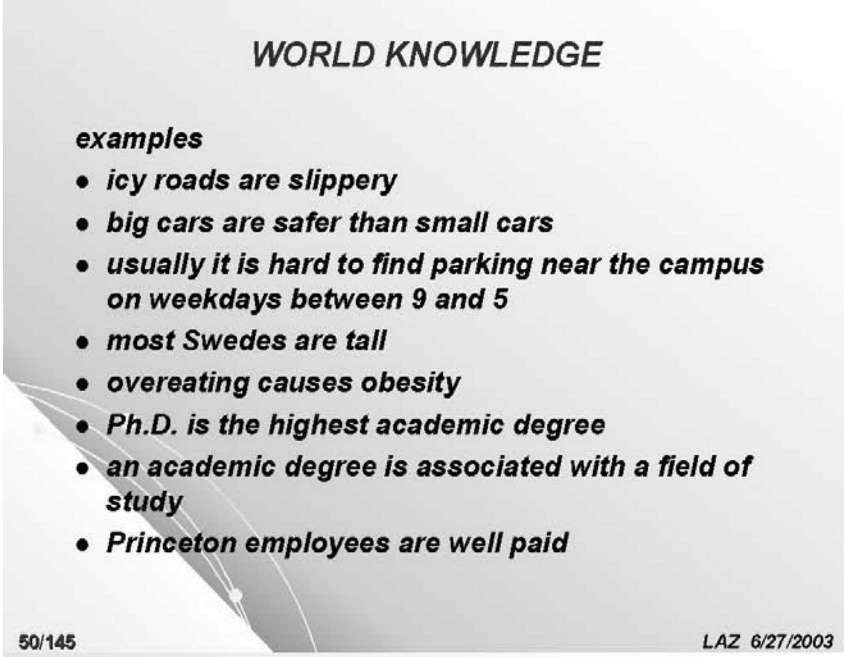
- P: usually Robert returns from work at about 6 pm
- P*: Prob {(Time(Return(Robert)) is 6 pm} is usually
- PF(p): Prob {X is A} is B
- X: Time (Return (Robert))
- A: 6 pm
- B: usually

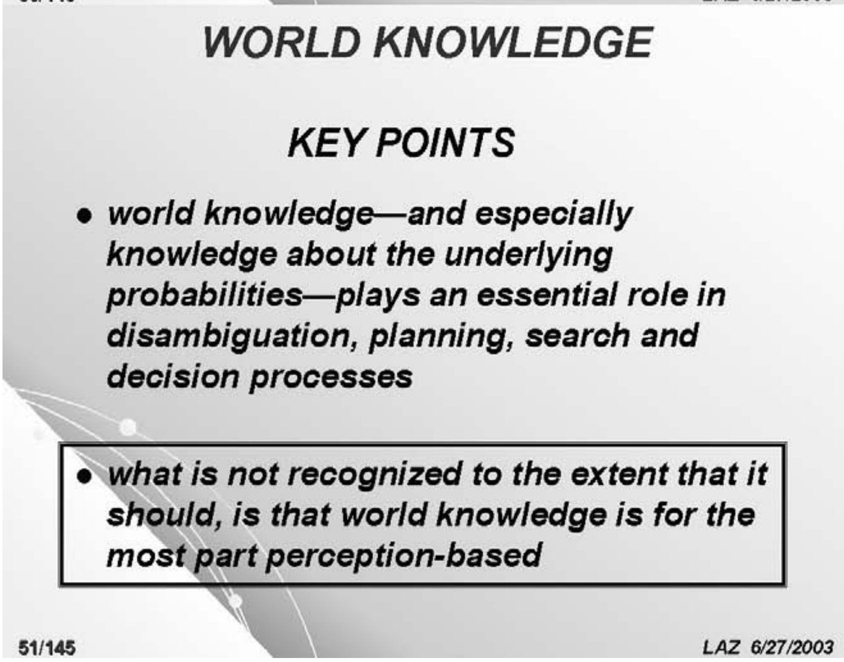
$p \in NL$
$p^* \in GCL$
$PF(p) \in PFL$

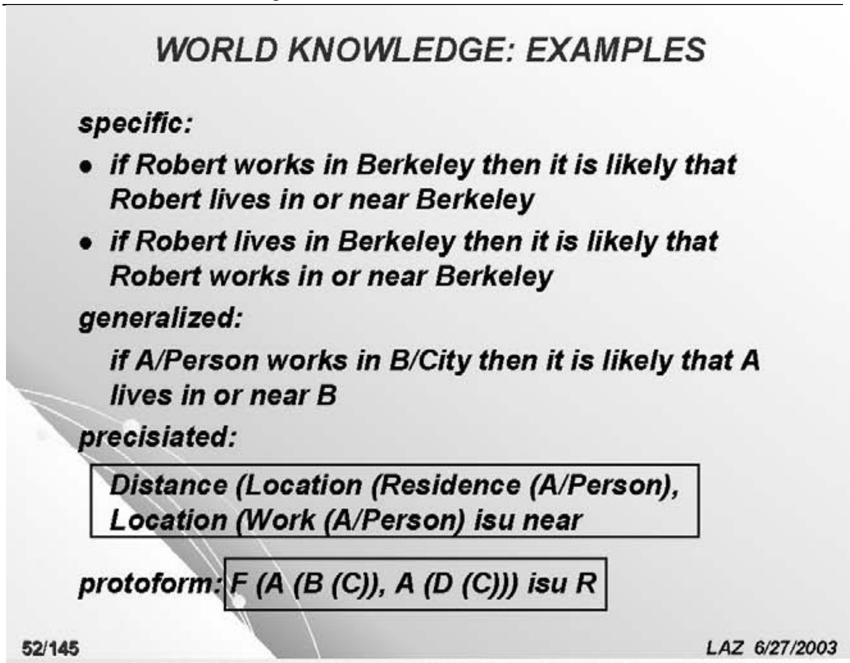
47/145 LAZ 6/27/2003

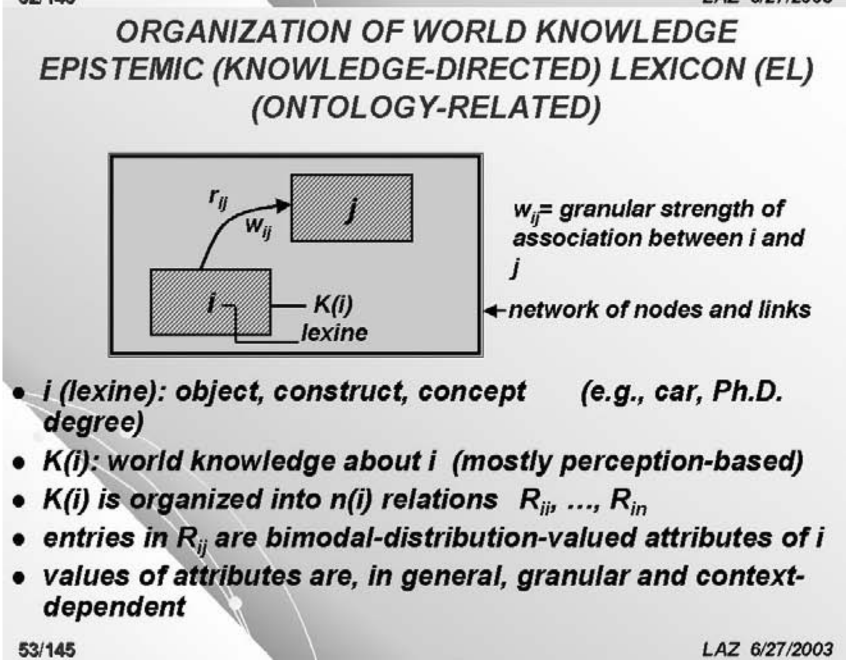

TRANSLATION FROM NL TO PFL
examples
Eva is young → A (B) is C
Age Eva young
Eva is much younger than Pat → (A (B), A (C)) is R
Age Eva Age Pat much younger
usually Robert returns from work at about 6pm →
Prob {A is B} is C
usually
about 6 pm
Time (Robert returns from work)
48/145
LAZ 6/27/2003

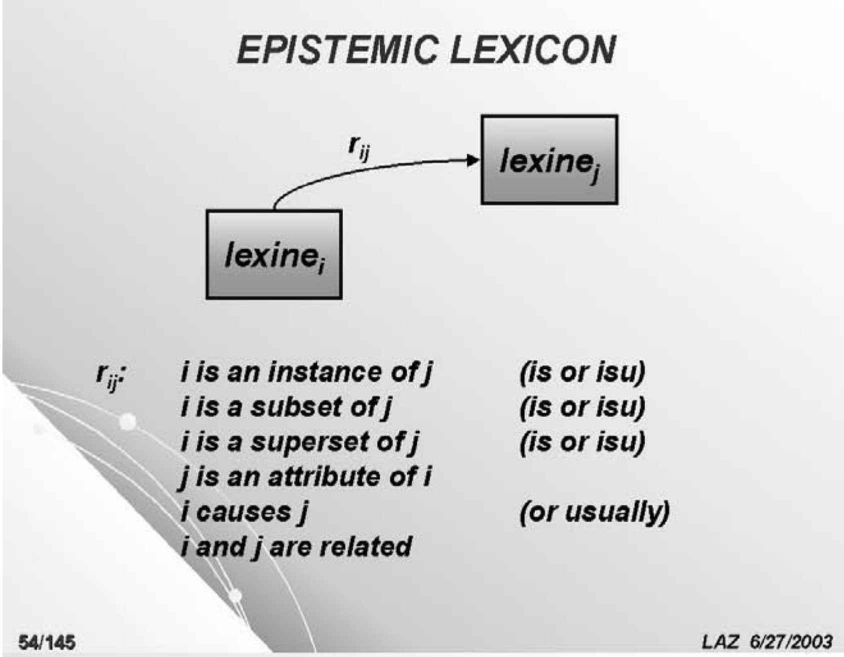

BASIC STRUCTURE OF PNL
NL
GCL
PFL
p
precisiation
p*
GC(p)
p**
PF(p)
precisiation (a)
abstraction (b)
WKDB
DDB
world knowledge database
deduction database
•In PNL, deduction=generalized constraint propagation
DDB: deduction database=collection of protoformal rules governing generalized constraint propagation
WKDB: PNL-based
LAZ 6/27/2003

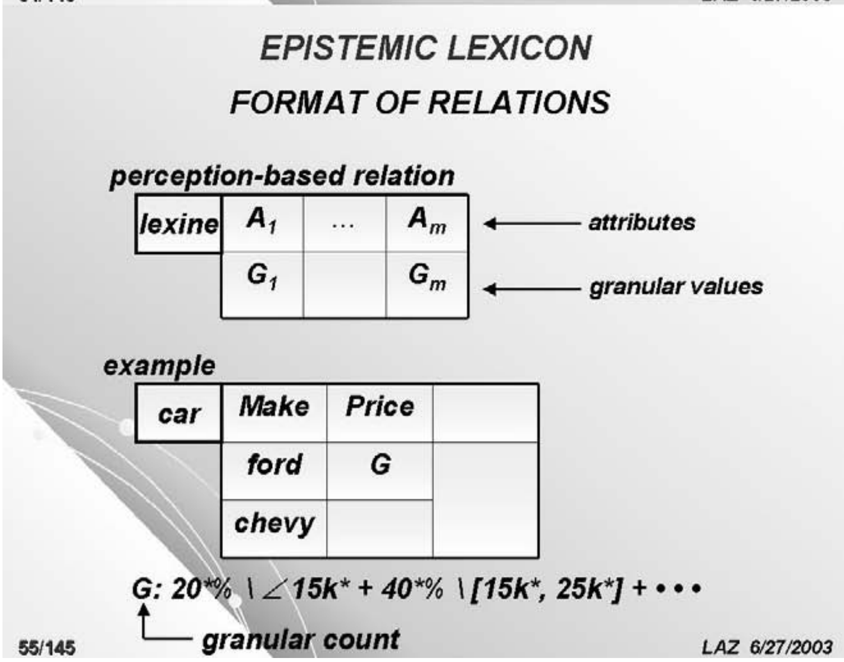

WORLD KNOWLEDGE
examples
icy roads are slippery
big cars are safer than small cars
usually it is hard to find parking near the campus on weekdays between 9 and 5
most Swedes are tall
overeating causes obesity
Ph.D. is the highest academic degree
an academic degree is associated with a field of study
Princeton employees are well paid
50/145
LAZ 6/27/2003

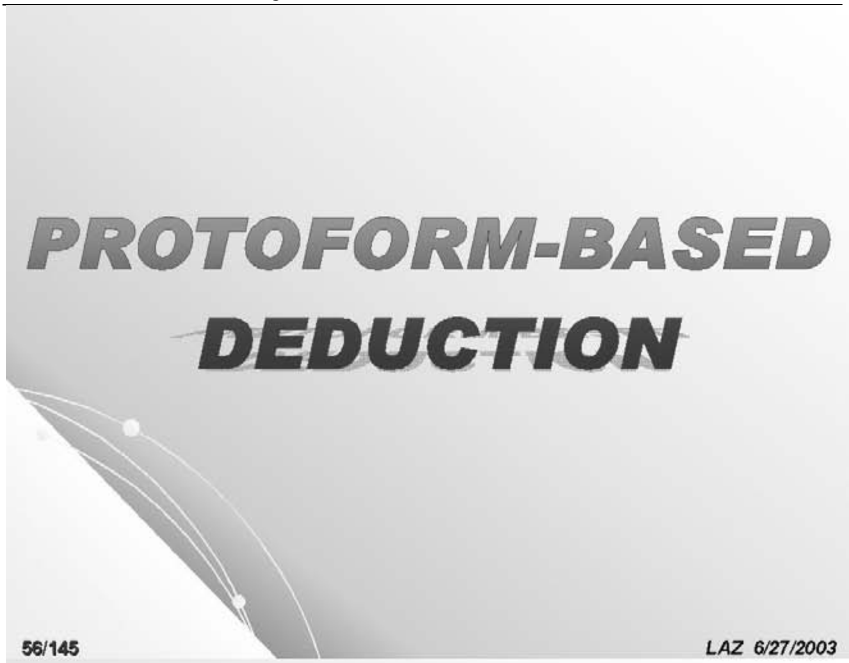

WORLD KNOWLEDGE
KEY POINTS
world knowledge—and especially knowledge about the underlying probabilities—plays an essential role in disambiguation, planning, search and decision processes
what is not recognized to the extent that it should, is that world knowledge is for the most part perception-based
51/145
LAZ 6/27/2003

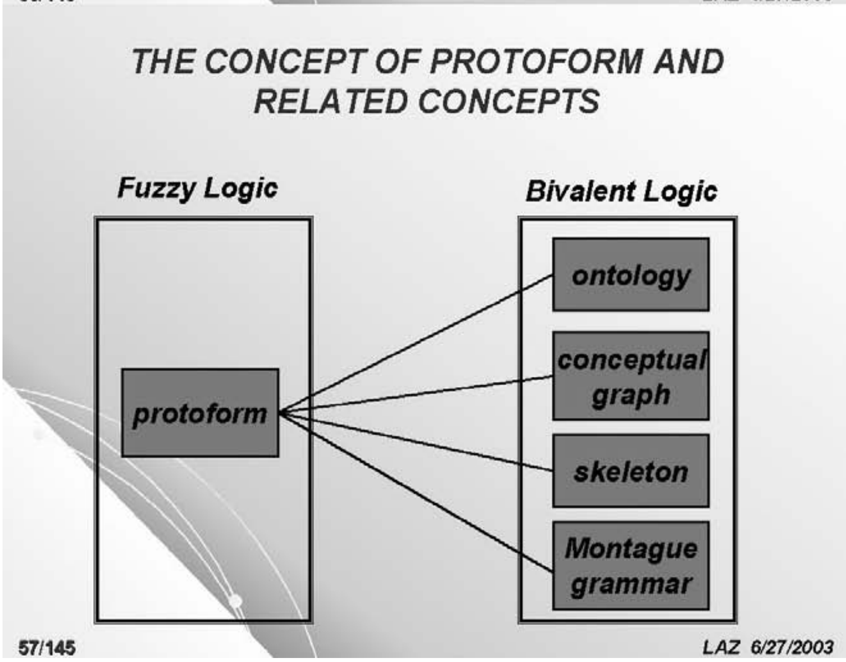

WORLD KNOWLEDGE: EXAMPLES
specific:
• if Robert works in Berkeley then it is likely that Robert lives in or near Berkeley
• if Robert lives in Berkeley then it is likely that Robert works in or near Berkeley
generalized:
if A/Person works in B/City then it is likely that A lives in or near B
precisiated:
Distance (Location (Residence (A/Person), Location (Work (A/Person) isu near
protoform: F (A (B (C)), A (D (C))) isu R
52/145
LAZ 6/27/2003

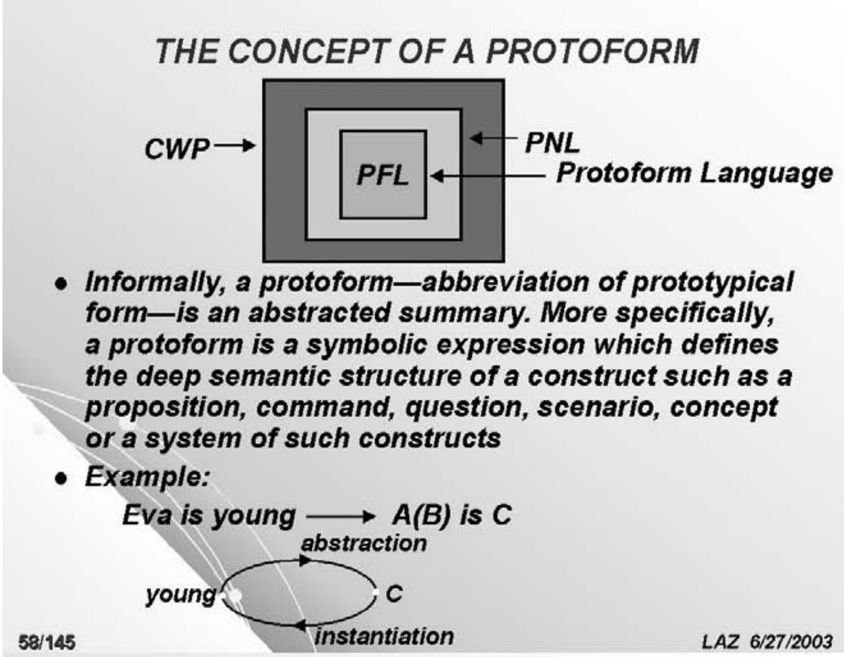

ORGANIZATION OF WORLD KNOWLEDGE
EPISTEMIC (KNOWLEDGE-DIRECTED) LEXICON (EL)
(ONTOLOGY-RELATED)
r_{ij}
w_{ij}
j
i
K(i)
lexine
w_{ij}= granular strength of association between i and j
network of nodes and links
• i (lexine): object, construct, concept (e.g., car, Ph.D. degree)
• K(i): world knowledge about i (mostly perception-based)
• K(i) is organized into n(i) relations R_{ii}, ..., R_{in}
• entries in R_{ij} are bimodal-distribution-valued attributes of i
• values of attributes are, in general, granular and context-dependent
53/145
LAZ 6/27/2003

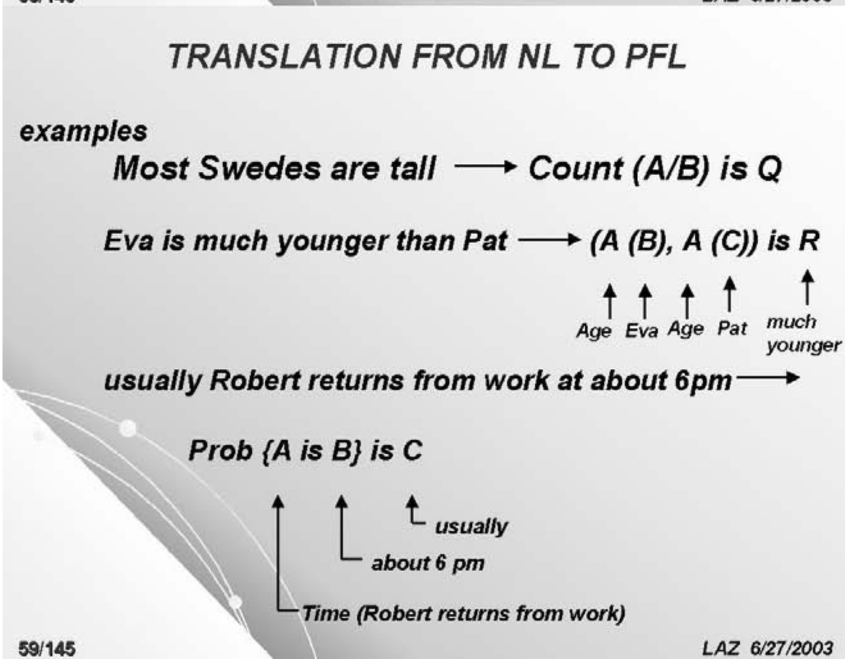

EPISTEMIC LEXICON
r_{ij}
$lexine_j$
$lexine_i$
r_{ij}: i is an instance of j (is or isu)
i is a subset of j (is or isu)
i is a superset of j (is or isu)
j is an attribute of i
i causes j (or usually)
i and j are related
54/145
LAZ 6/27/2003

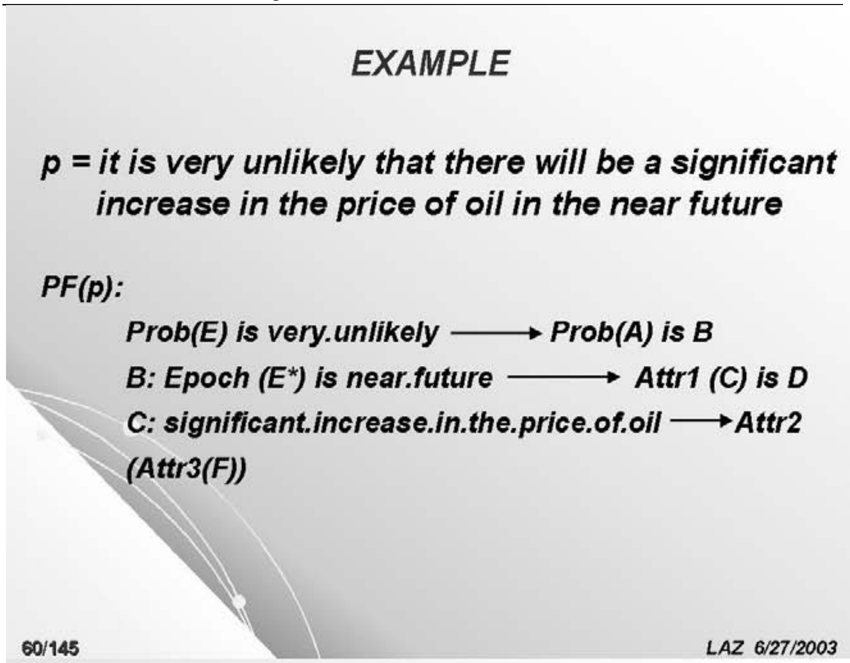

EPISTEMIC LEXICON
FORMAT OF RELATIONS
perception-based relation
lexine
A_1
…
A_m
G_1
G_m
attributes
granular values
example
car
Make
Price
ford
G
chevy
G: 20*% \ ∠15k* + 40*% \ [15k*, 25k*] + • • •
granular count
55/145
LAZ 6/27/2003

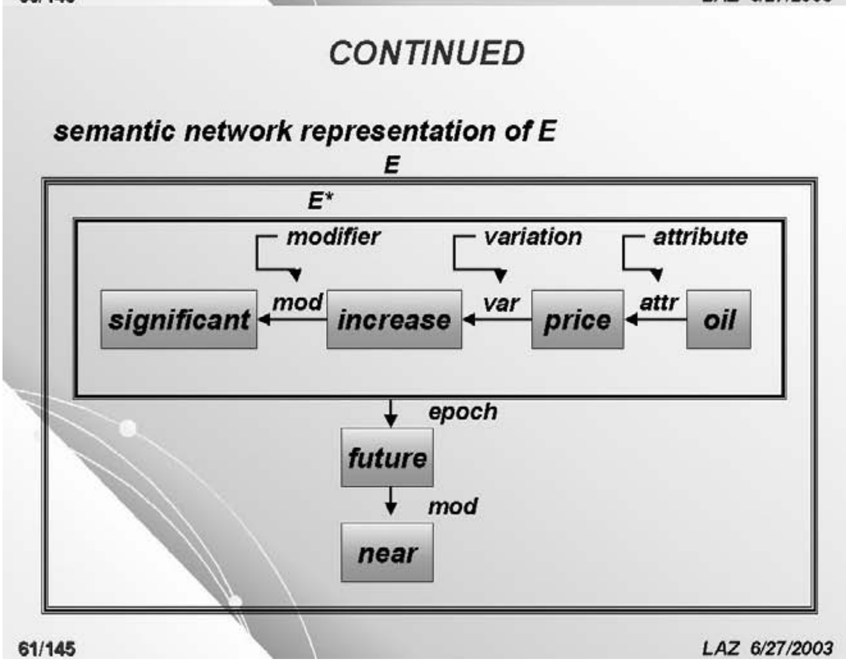

PROTOFORM-BASED
DEDUCTION
56/145
LAZ 6/27/2003

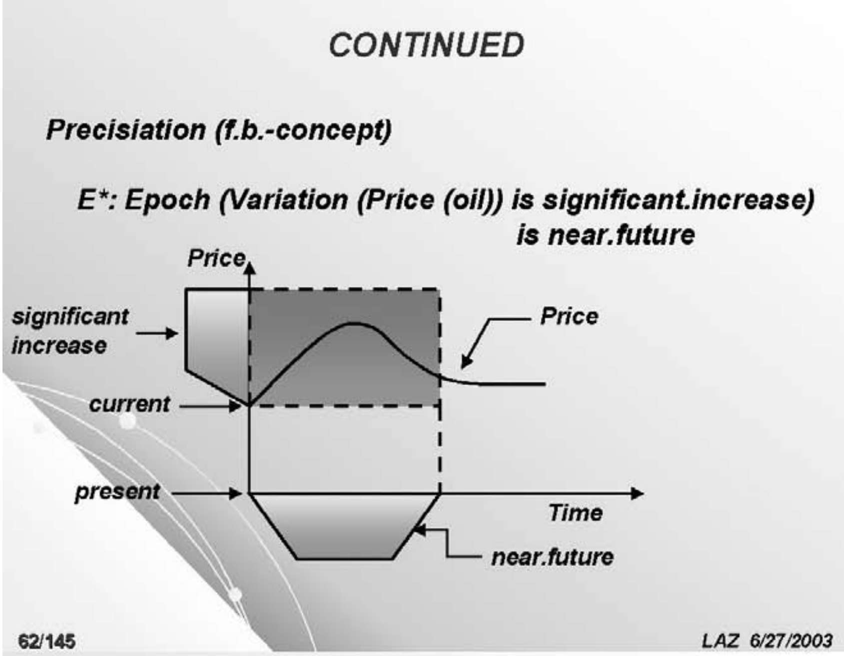

THE CONCEPT OF PROTOFORM AND RELATED CONCEPTS
Fuzzy Logic
Bivalent Logic
protoform
ontology
conceptual graph
skeleton
Montague grammar
57/145
LAZ 6/27/2003

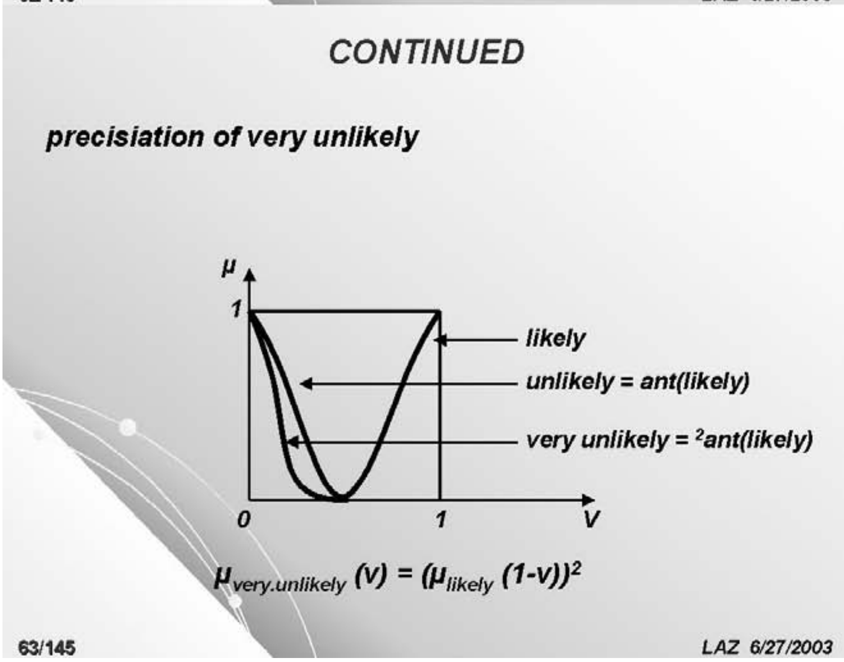

THE CONCEPT OF A PROTOFORM
CWP
PNL
PFL
Protoform Language
• Informally, a protoform—abbreviation of prototypical form—is an abstracted summary. More specifically, a protoform is a symbolic expression which defines the deep semantic structure of a construct such as a proposition, command, question, scenario, concept or a system of such constructs
• Example:
Eva is young → A(B) is C
abstraction
young
C
instantiation
58/145
LAZ 6/27/2003

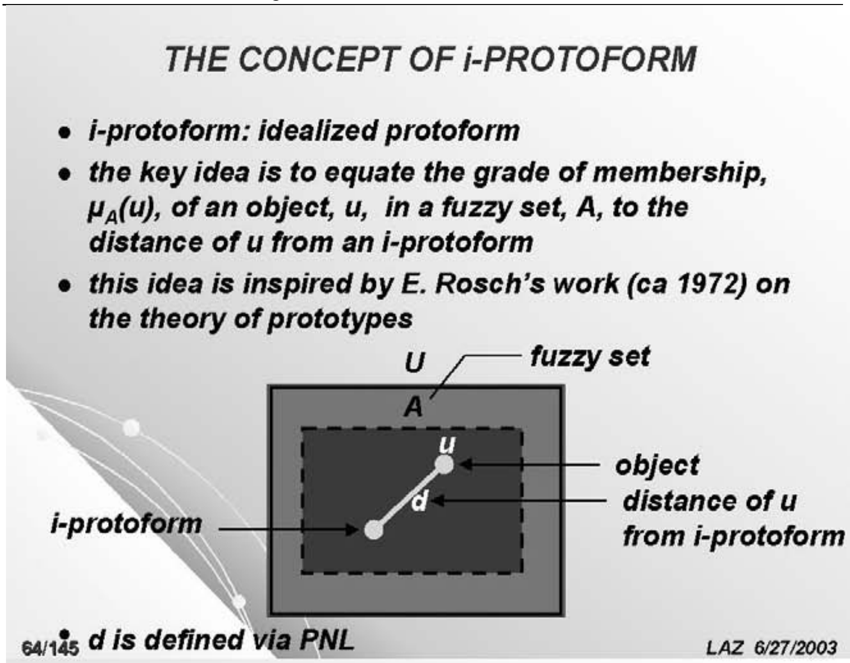

TRANSLATION FROM NL TO PFL
examples
Most Swedes are tall → Count (A/B) is Q
Eva is much younger than Pat → (A (B), A (C)) is R
Age
Eva
Age
Pat
much younger
usually Robert returns from work at about 6pm →
Prob {A is B} is C
usually
about 6 pm
Time (Robert returns from work)
59/145
LAZ 6/27/2003

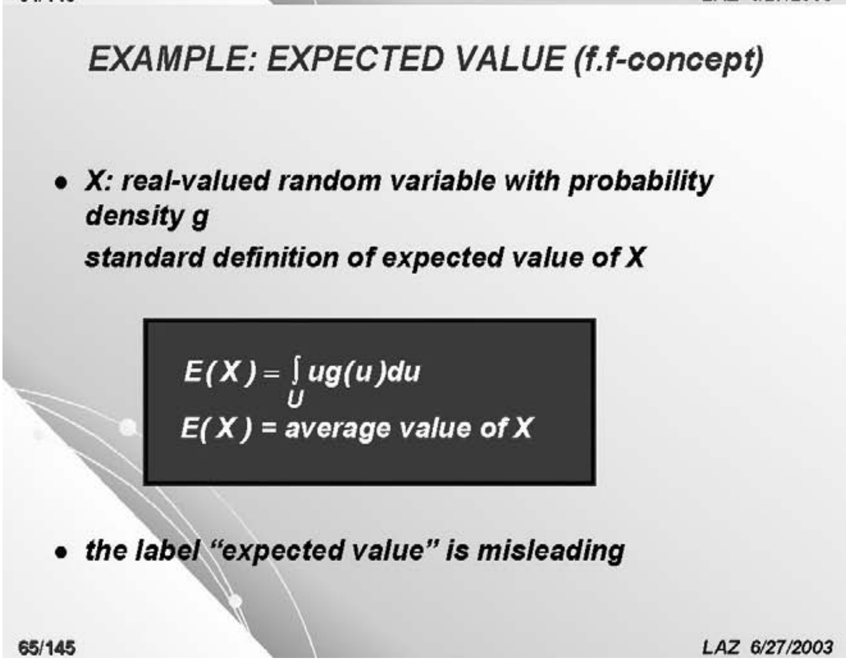

EXAMPLE
p = it is very unlikely that there will be a significant increase in the price of oil in the near future
PF(p):
Prob(E) is very.unlikely → Prob(A) is B
B: Epoch (E*) is near.future → Attr1 (C) is D
C: significant.increase.in.the.price.of.oil → Attr2 (Attr3(F))
60/145
LAZ 6/27/2003

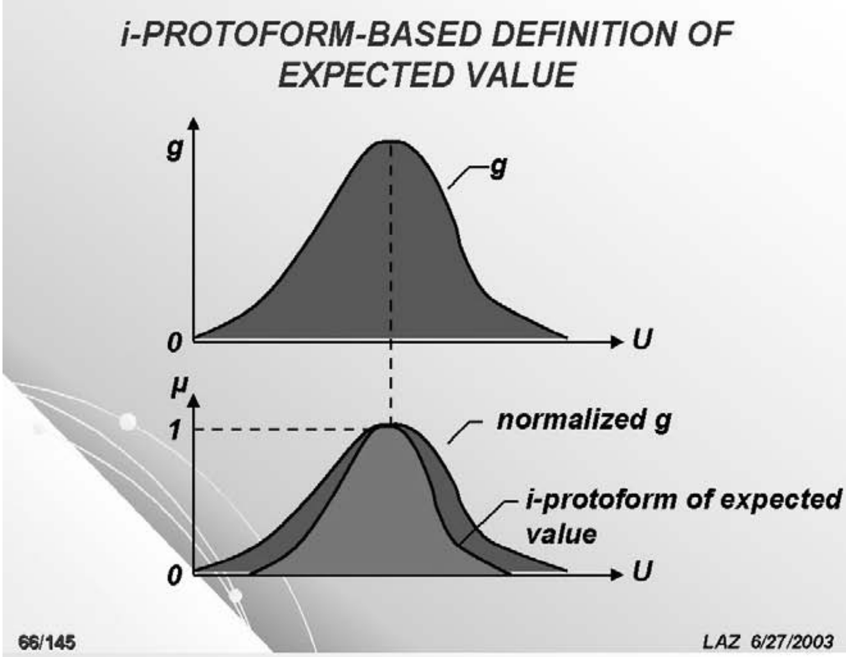

CONTINUED
semantic network representation of E
E
E*
modifier
variation
attribute
significant
mod
increase
var
price
attr
oil
epoch
future
mod
near
61/145
LAZ 6/27/2003

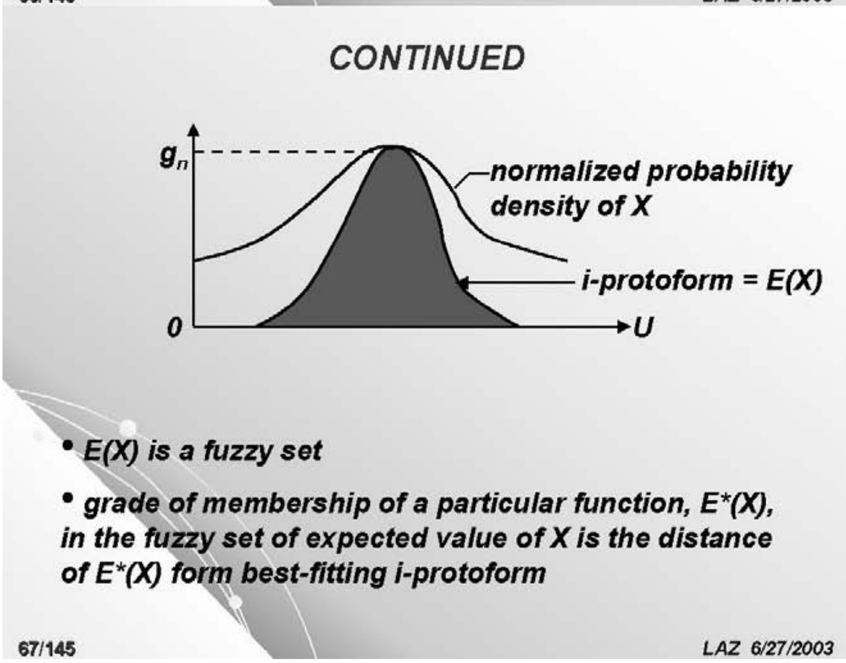

CONTINUED
Precisiation (f.b.-concept)
E*: Epoch (Variation (Price (oil)) is significant.increase) is near.future
Price
significant increase
Price
current
present
Time
near.future
62/145
LAZ 6/27/2003

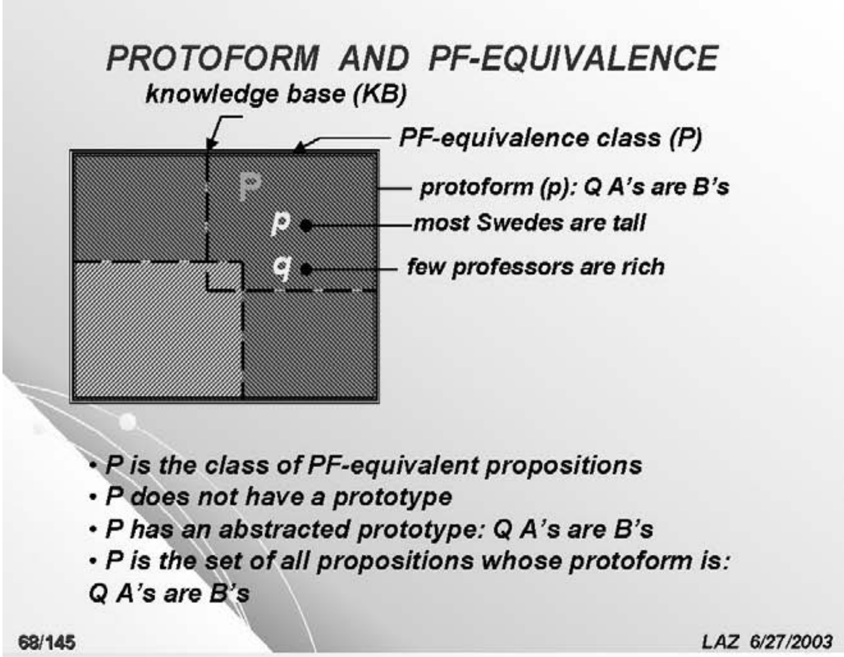

CONTINUED
precisiation of very unlikely
μ
1
likely
unlikely = ant(likely)
very unlikely = ²ant(likely)
0
1
V
μvery.unlikely (v) = (μlikely (1-v))²
63/145
LAZ 6/27/2003

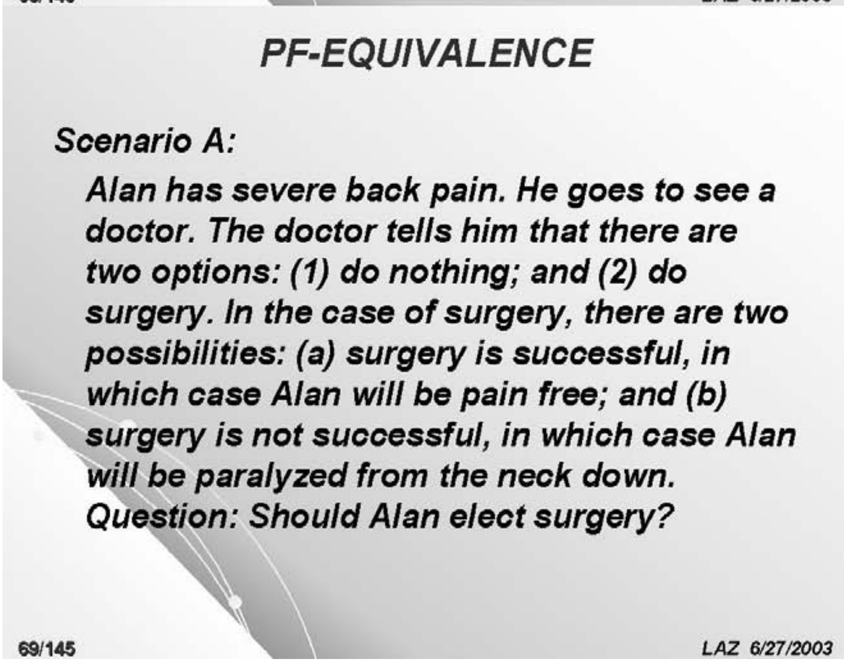

THE CONCEPT OF i-PROTOFORM
• i-protoform: idealized protoform
• the key idea is to equate the grade of membership, $\mu_A(u)$, of an object, u, in a fuzzy set, A, to the distance of u from an i-protoform
• this idea is inspired by E. Rosch's work (ca 1972) on the theory of prototypes
U
fuzzy set
A
u
object
d
distance of u from i-protoform
i-protoform
• d is defined via PNL
64/145
LAZ 6/27/2003

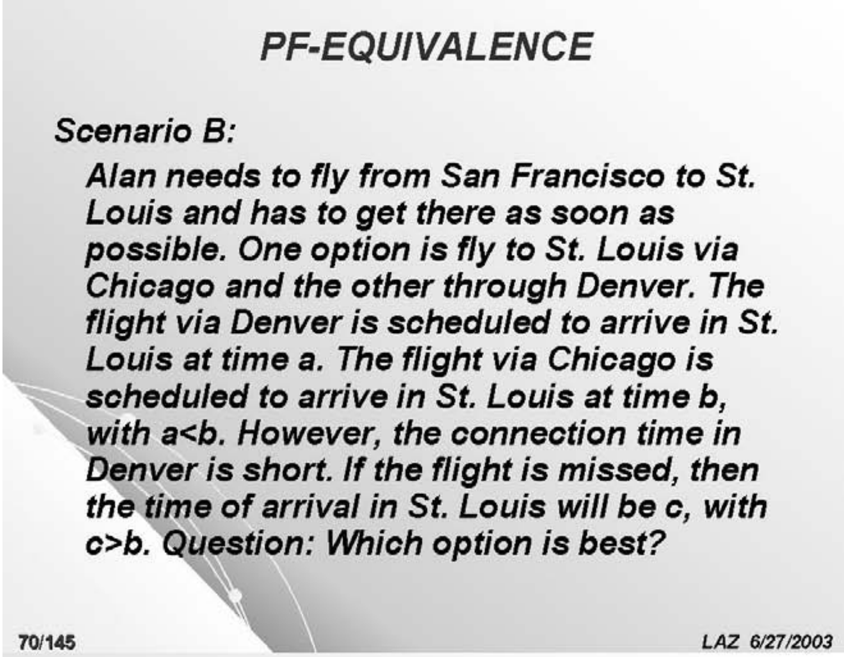

EXAMPLE: EXPECTED VALUE (f.f-concept)
• X: real-valued random variable with probability density g
standard definition of expected value of X
$E(X) = \int_U ug(u)du$
E(X) = average value of X
• the label "expected value" is misleading
65/145
LAZ 6/27/2003

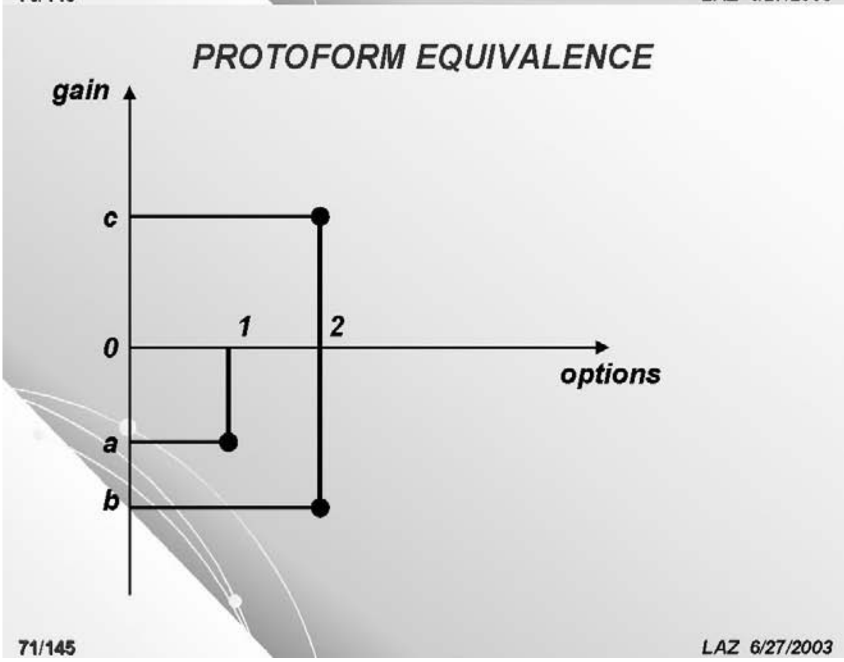

i-PROTOFORM-BASED DEFINITION OF EXPECTED VALUE
g
g
0
U
μ
1
normalized g
i-protoform of expected value
0
U
66/145
LAZ 6/27/2003

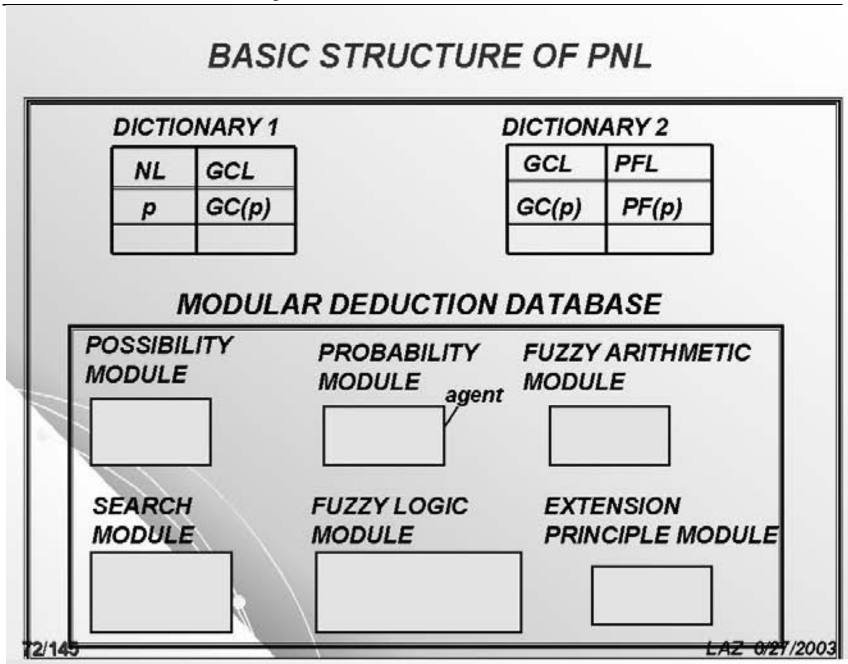

CONTINUED
g_n
normalized probability density of X
i-protoform = E(X)
0
U
• E(X) is a fuzzy set
• grade of membership of a particular function, E*(X), in the fuzzy set of expected value of X is the distance of E*(X) form best-fitting i-protoform
67/145
LAZ 6/27/2003

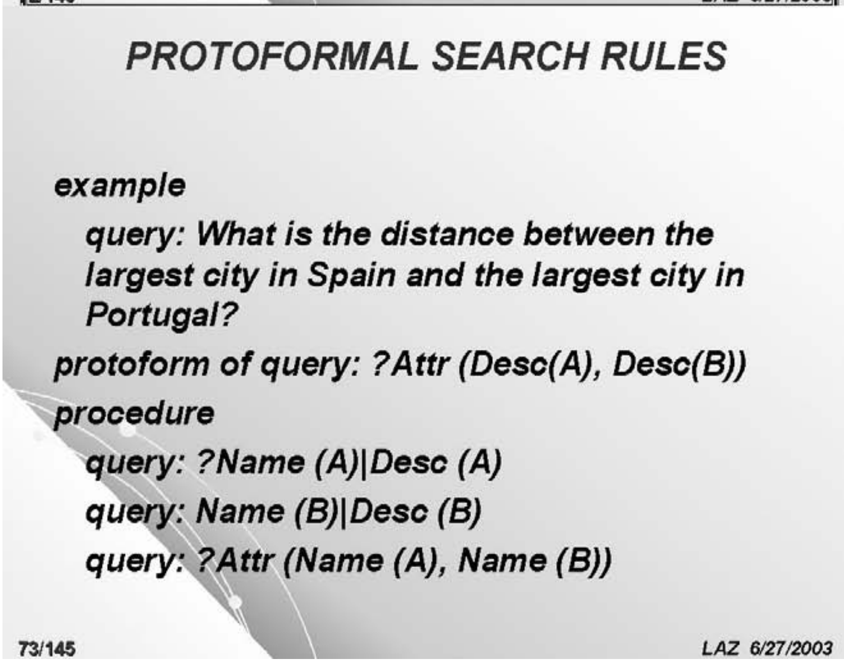

PROTOFORM AND PF-EQUIVALENCE
knowledge base (KB)
PF-equivalence class (P)
P
protoform (p): Q A's are B's
p
most Swedes are tall
q
few professors are rich
• P is the class of PF-equivalent propositions
• P does not have a prototype
• P has an abstracted prototype: Q A's are B's
• P is the set of all propositions whose protoform is:
Q A's are B's
68/145
LAZ 6/27/2003

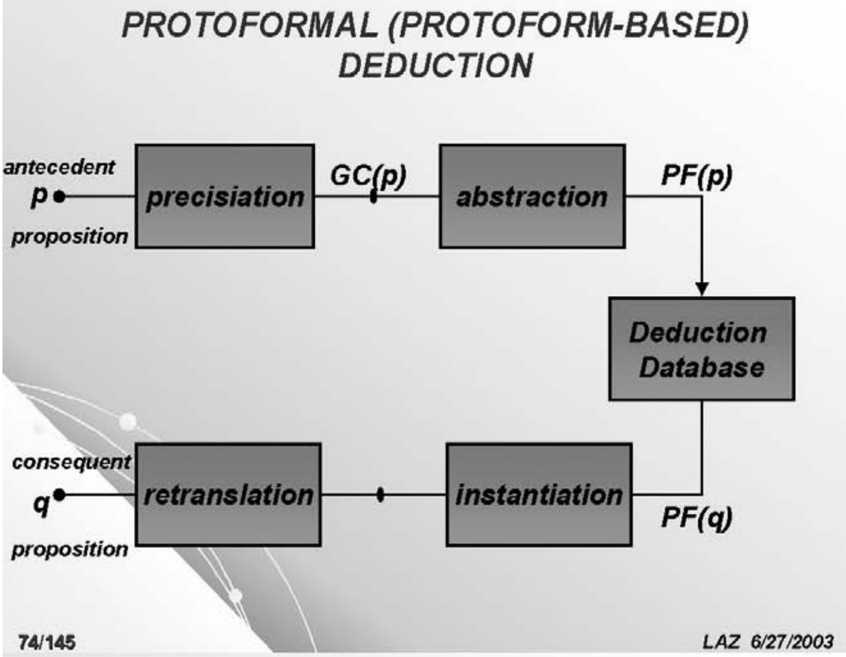

PF-EQUIVALENCE
Scenario A:
Alan has severe back pain. He goes to see a doctor. The doctor tells him that there are two options: (1) do nothing; and (2) do surgery. In the case of surgery, there are two possibilities: (a) surgery is successful, in which case Alan will be pain free; and (b) surgery is not successful, in which case Alan will be paralyzed from the neck down.
Question: Should Alan elect surgery?
69/145
LAZ 6/27/2003

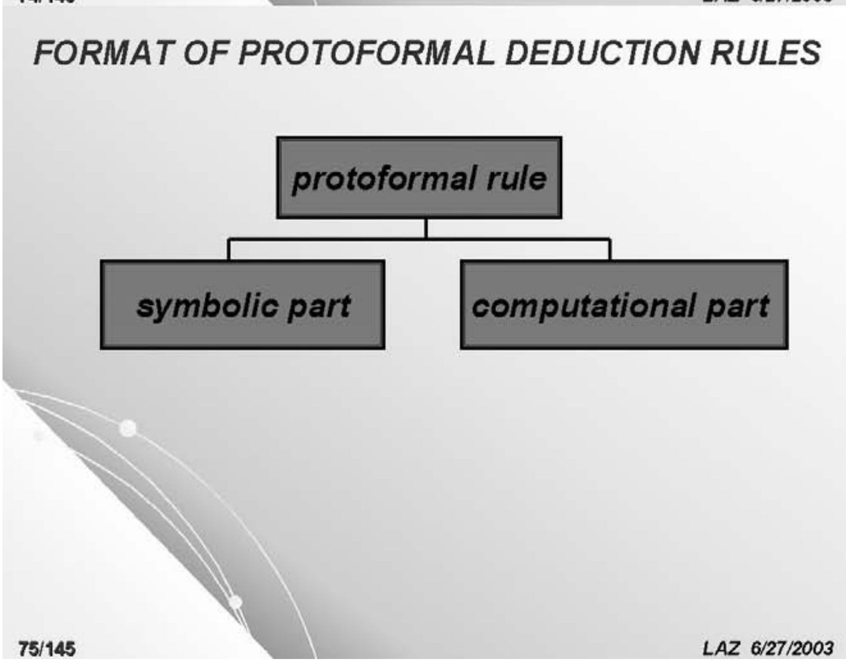

PF-EQUIVALENCE
Scenario B:
Alan needs to fly from San Francisco to St. Louis and has to get there as soon as possible. One option is fly to St. Louis via Chicago and the other through Denver. The flight via Denver is scheduled to arrive in St. Louis at time a. The flight via Chicago is scheduled to arrive in St. Louis at time b, with a<b. However, the connection time in Denver is short. If the flight is missed, then the time of arrival in St. Louis will be c, with c>b. Question: Which option is best?
70/145
LAZ 6/27/2003

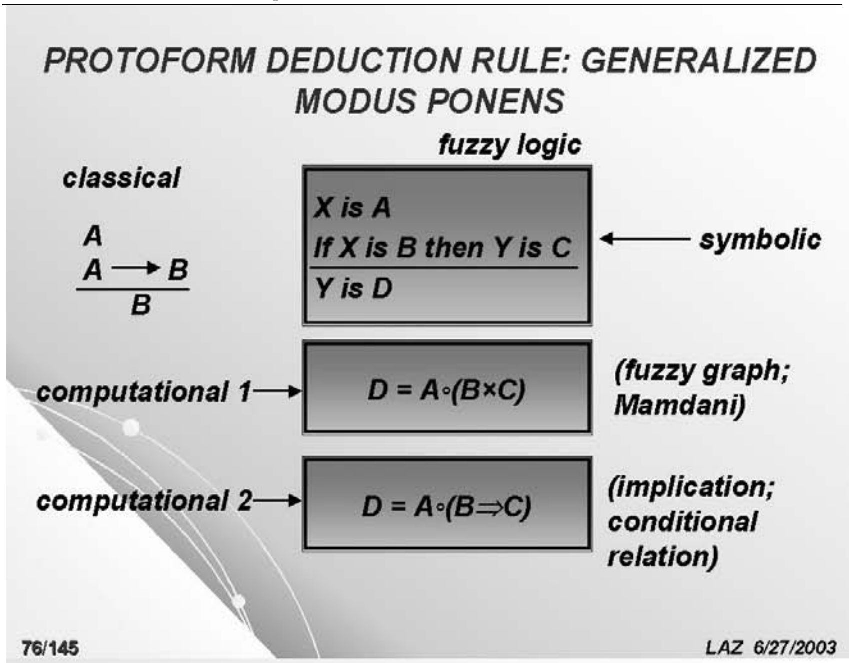

PROTOFORM EQUIVALENCE
gain
c
1
2
0
options
a
b
71/145
LAZ 6/27/2003

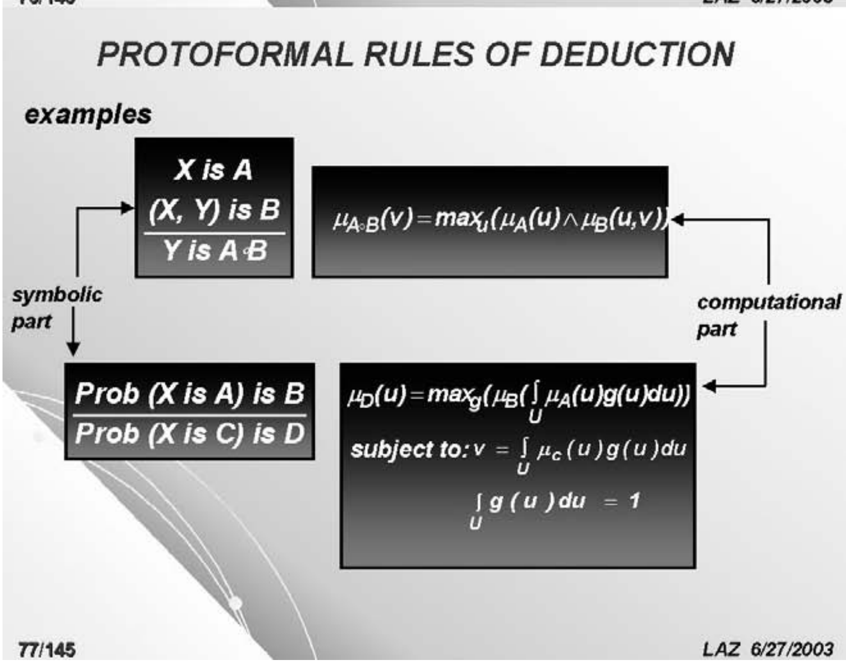

BASIC STRUCTURE OF PNL
DICTIONARY 1
NL
GCL
p
GC(p)
DICTIONARY 2
GCL
PFL
GC(p)
PF(p)
MODULAR DEDUCTION DATABASE
POSSIBILITY MODULE
PROBABILITY MODULE
agent
FUZZY ARITHMETIC MODULE
SEARCH MODULE
FUZZY LOGIC MODULE
EXTENSION PRINCIPLE MODULE
72/145
LAZ 6/27/2003

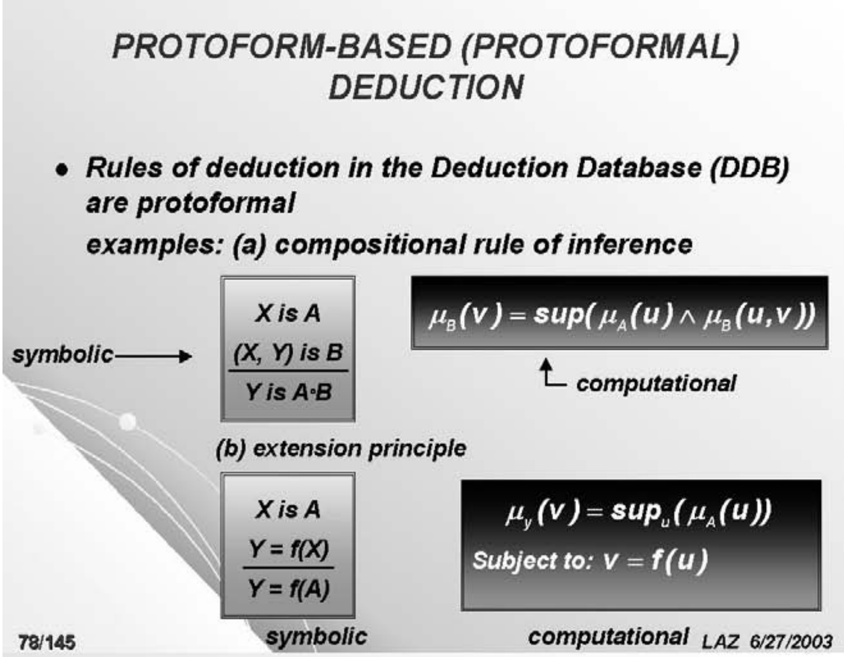

PROTOFORMAL SEARCH RULES
example
query: What is the distance between the largest city in Spain and the largest city in Portugal?
protoform of query: ?Attr (Desc(A), Desc(B))
procedure
query: ?Name (A)|Desc (A)
query: Name (B)|Desc (B)
query: ?Attr (Name (A), Name (B))
73/145
LAZ 6/27/2003

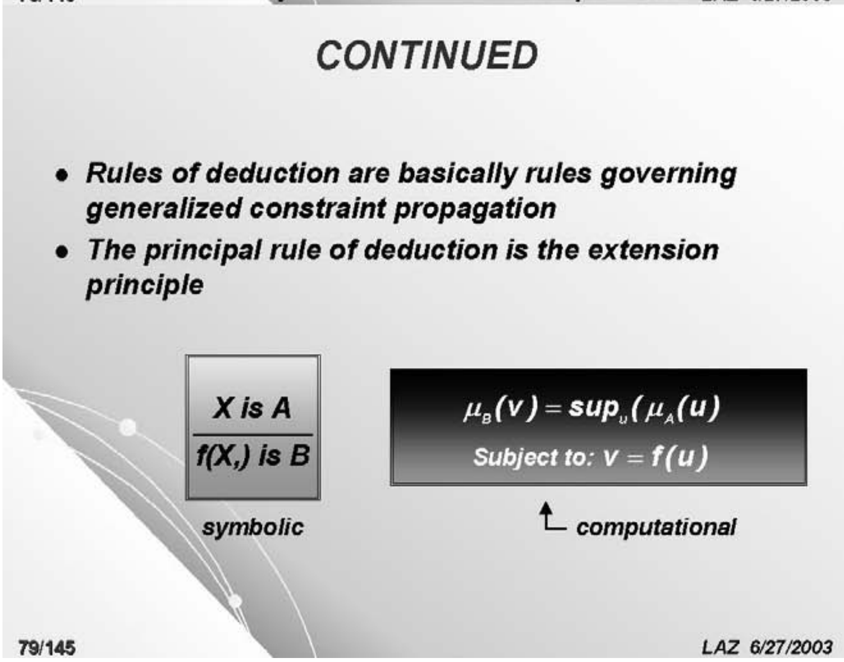

PROTOFORMAL (PROTOFORM-BASED) DEDUCTION
antecedent
p
proposition
precisiation
GC(p)
abstraction
PF(p)
Deduction Database
consequent
q
proposition
retranslation
instantiation
PF(q)
74/145
LAZ 6/27/2003

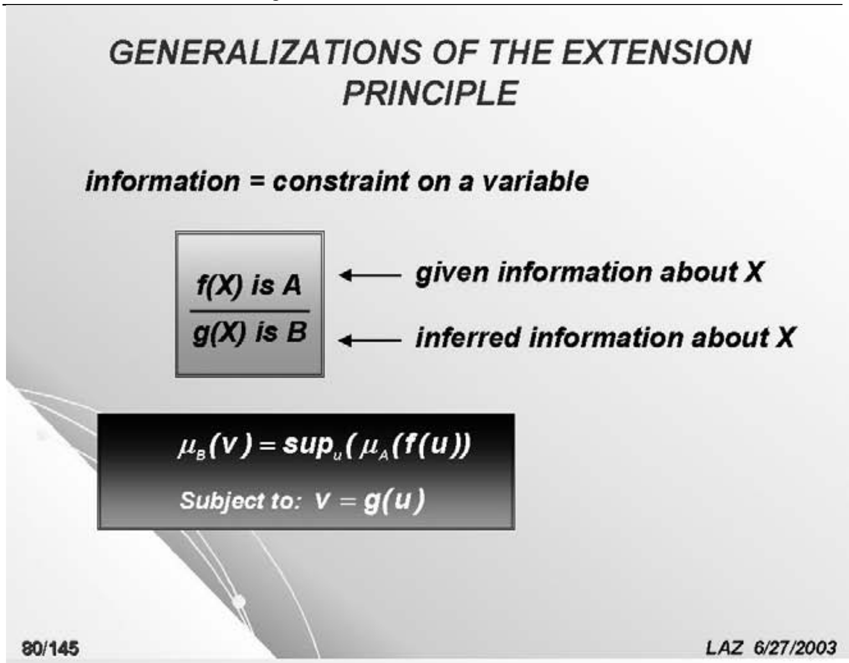

FORMAT OF PROTOFORMAL DEDUCTION RULES
protoformal rule
symbolic part
computational part
75/145
LAZ 6/27/2003

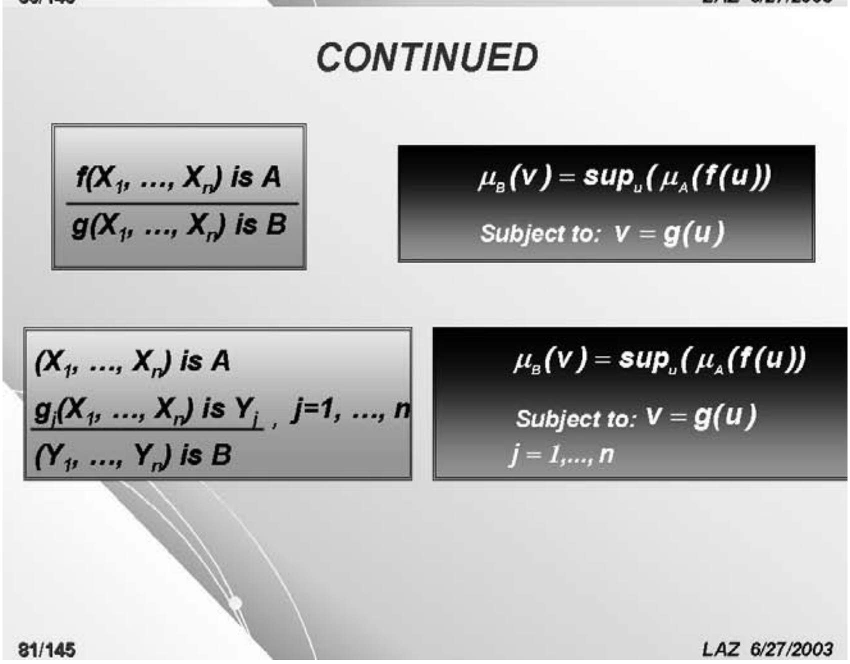

PROTOFORM DEDUCTION RULE: GENERALIZED MODUS PONENS
fuzzy logic
classical
A
A → B
B
X is A
If X is B then Y is C
Y is D
symbolic
computational 1
D = A∘(B×C)
(fuzzy graph; Mamdani)
computational 2
D = A∘(B⇒C)
(implication; conditional relation)
76/145
LAZ 6/27/2003

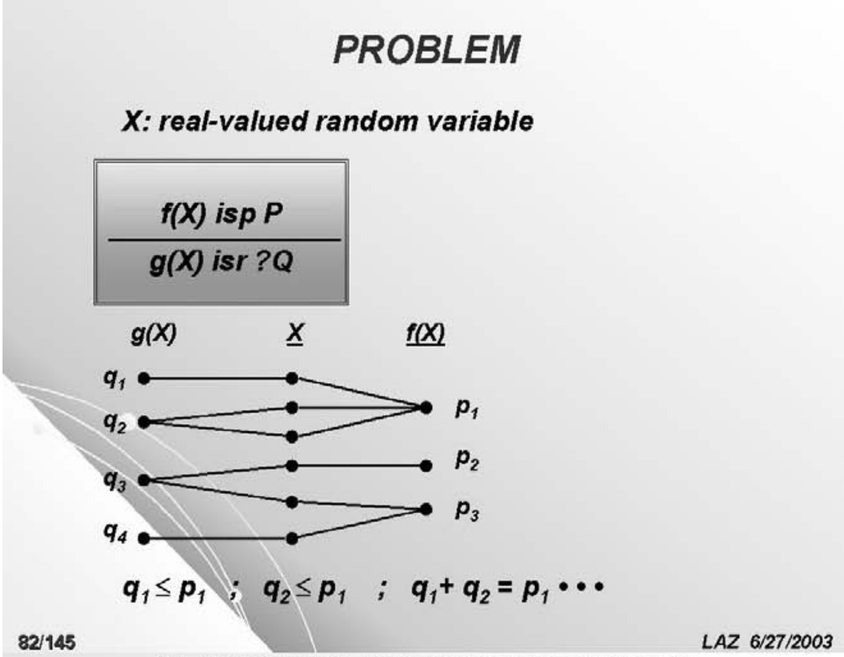

PROTOFORMAL RULES OF DEDUCTION
examples
X is A
(X, Y) is B
Y is A∘B
μA∘B(v) = maxu(μA(u) ∧ μB(u,v))
symbolic part
computational part
Prob (X is A) is B
Prob (X is C) is D
μD(u) = maxg(μB(∫U μA(u)g(u)du))
subject to: v = ∫U μc(u)g(u)du
∫U g(u)du = 1
77/145
LAZ 6/27/2003

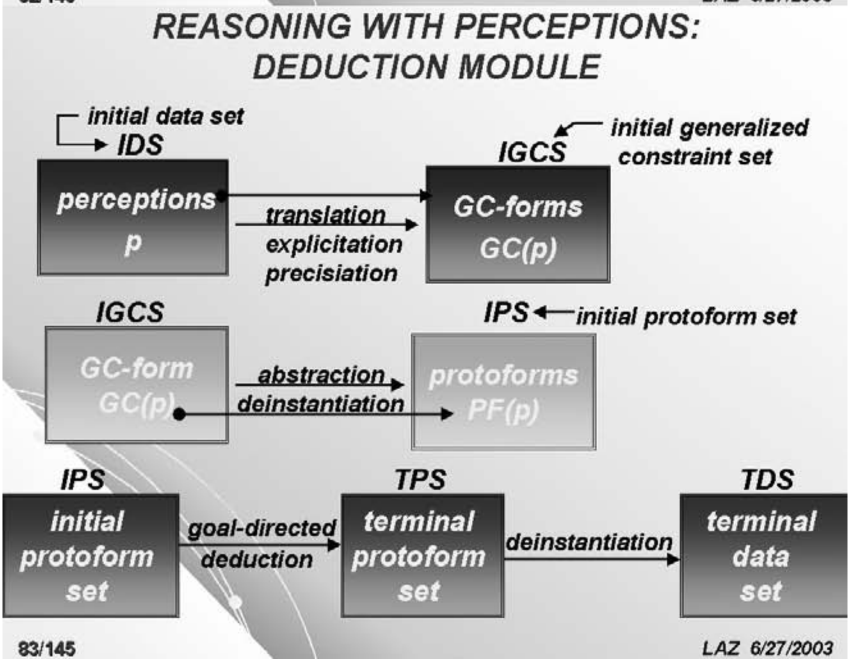

PROTOFORM-BASED (PROTOFORMAL) DEDUCTION
• Rules of deduction in the Deduction Database (DDB) are protoformal
examples: (a) compositional rule of inference
symbolic
X is A
(X, Y) is B
Y is A∘B
$\mu_B(v) = sup(\mu_A(u) \wedge \mu_B(u,v))$
computational
(b) extension principle
X is A
Y = f(X)
Y = f(A)
$\mu_y(v) = sup_u(\mu_A(u))$
Subject to: $v = f(u)$
symbolic
computational
78/145
LAZ 6/27/2003


CONTINUED
• Rules of deduction are basically rules governing generalized constraint propagation
• The principal rule of deduction is the extension principle
X is A
f(X,) is B
$\mu_B(v) = sup_u(\mu_A(u)$
Subject to: $v = f(u)$
symbolic
computational
79/145
LAZ 6/27/2003


GENERALIZATIONS OF THE EXTENSION PRINCIPLE
information = constraint on a variable
f(X) is A
g(X) is B
given information about X
inferred information about X
$\mu_B(v) = sup_u(\mu_A(f(u))$
Subject to: $v = g(u)$
80/145
LAZ 6/27/2003


CONTINUED
$f(X_1, \ldots, X_n)$ is A
$g(X_1, \ldots, X_n)$ is B
$\mu_B(v) = sup_u(\mu_A(f(u))$
Subject to: $v = g(u)$
$(X_1, \ldots, X_n)$ is A
$g_j(X_1, \ldots, X_n)$ is Y_j , j=1, ..., n
$(Y_1, \ldots, Y_n)$ is B
$\mu_B(v) = sup_u(\mu_A(f(u))$
Subject to: $v = g(u)$
$j = 1,\ldots, n$
81/145
LAZ 6/27/2003

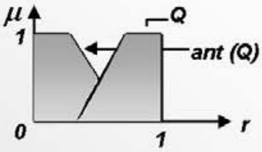

PROBLEM
X: real-valued random variable
f(X) isp P
g(X) isr ?Q
g(X)
X
f(X)
q1
q2
q3
q4
p1
p2
p3
q1 ≤ p1 ; q2 ≤ p1 ; q1 + q2 = p1 • • •
82/145
LAZ 6/27/2003


REASONING WITH PERCEPTIONS: DEDUCTION MODULE
initial data set
IDS
IGCS
initial generalized constraint set
perceptions p
translation
explicitation
precisiation
GC-forms GC(p)
IGCS
IPS
initial protoform set
GC-form GC(p)
abstraction
deinstantiation
protoforms PF(p)
IPS
TPS
TDS
initial protoform set
goal-directed deduction
terminal protoform set
deinstantiation
terminal data set
83/145
LAZ 6/27/2003

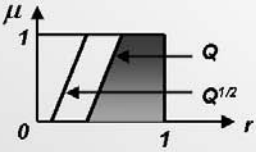
COUNT-AND MEASURE-RELATED RULES

crisp

$$\frac{Q \text{ A's are B's}}{\text{ant (Q) A's are not B's}}$$



$$\frac{Q \text{ A's are B's}}{Q^{1/2} \text{ A's are } {}^{2}\text{B's}}$$



$$\frac{\text{most Swedes are tall}}{\text{ave (height) Swedes is ?h}} \longrightarrow \frac{Q \text{ A's are B's}}{\text{ave (B|A) is ?C}}$$

$$\mu_{ave}(v) = sup_a \, \mu_Q\left(\frac{1}{N}\sum_i \mu_B(a_i)\right), \quad a = (a_1, \ldots, a_N)$$

$$v = \frac{1}{N}\left(\sum_i a_i\right)$$

84/145 LAZ 6/27/2003

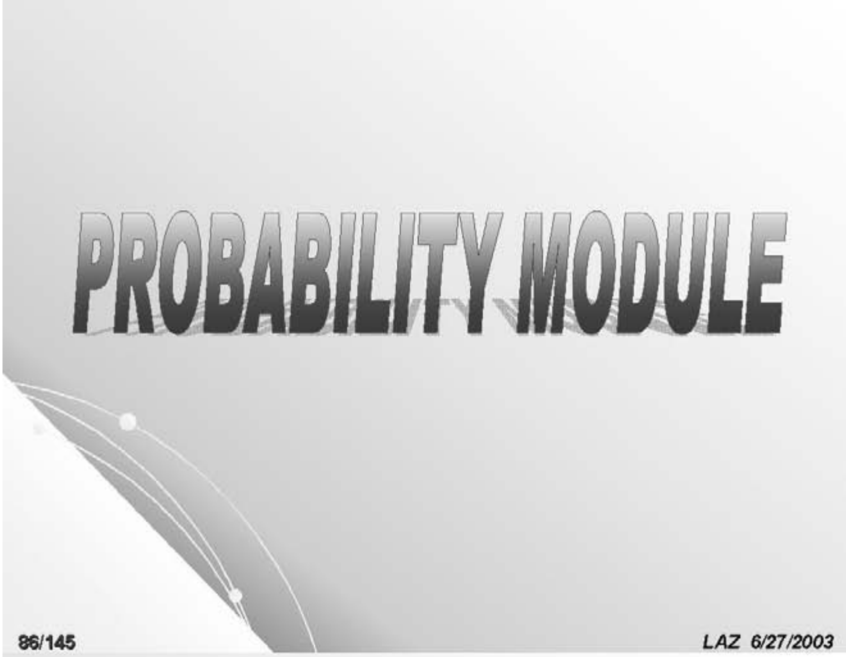
CONTINUED

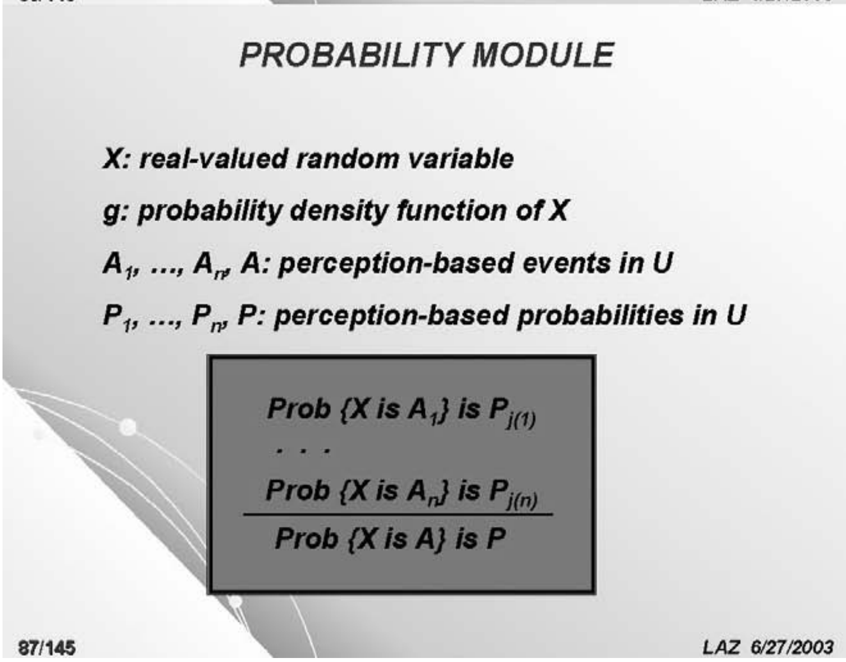
not(QA's are B's) $\longleftrightarrow$ (not Q) A's are B's

$$\frac{\begin{array}{l} Q_1 \text{ A's are B's} \\ Q_2 \text{ (A\&B)'s are C's} \end{array}}{Q_1 \, Q_2 \text{ A's are (B\&C)'s}}$$

$$\frac{\begin{array}{l} Q_1 \text{ A's are B's} \\ Q_2 \text{ A's are C's} \end{array}}{(Q_1 + Q_2 - 1) \text{ A's are (B\&C)'s}}$$

85/145 LAZ 6/27/2003


PROBABILITY MODULE
86/145
LAZ 6/27/2003


PROBABILITY MODULE
X: real-valued random variable
g: probability density function of X
$A_1, \ldots, A_n$, A: perception-based events in U
$P_1, \ldots, P_n$, P: perception-based probabilities in U
Prob {X is A_1} is $P_{j(1)}$
. . .
Prob {X is A_n} is $P_{j(n)}$
Prob {X is A} is P
87/145
LAZ 6/27/2003

CONTINUED

$$\mu_p(v) = sup_g(\mu_{P_1}(\int_U g(u)\mu_{A_1}(u)du) \wedge \cdots$$

$$\wedge \mu_{P_n}(\int_U g(u)\mu_{A_n}(u)du)$$

subject to:

$$v = \int_U g(u)\mu_A(u)du$$

88/145 LAZ 6/27/2003

PROBABILITY MODULE (CONTINUED)

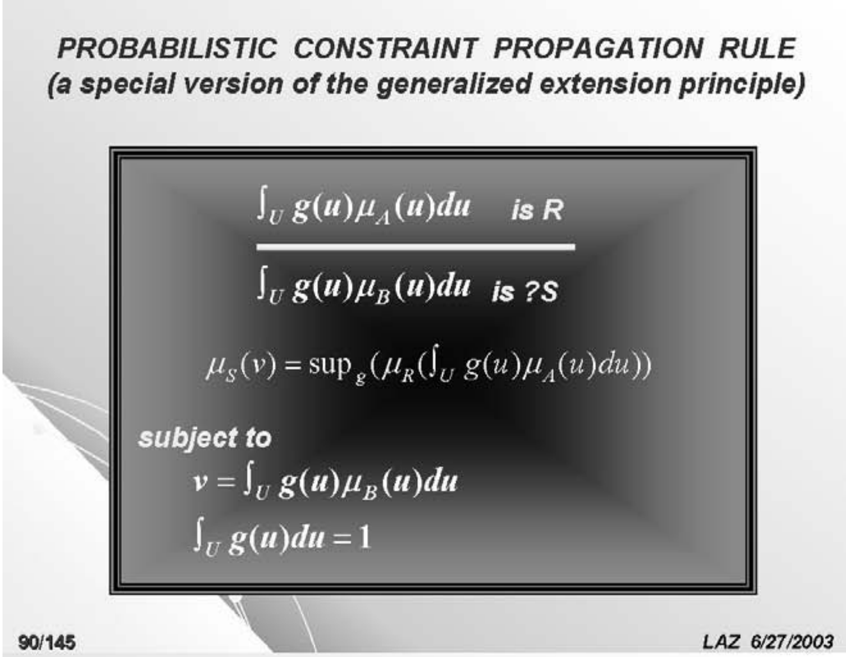
$$\frac{\begin{array}{c}X \text{ isp } P\\ Y = f(X)\end{array}}{Y \text{ isp } f(P)}$$

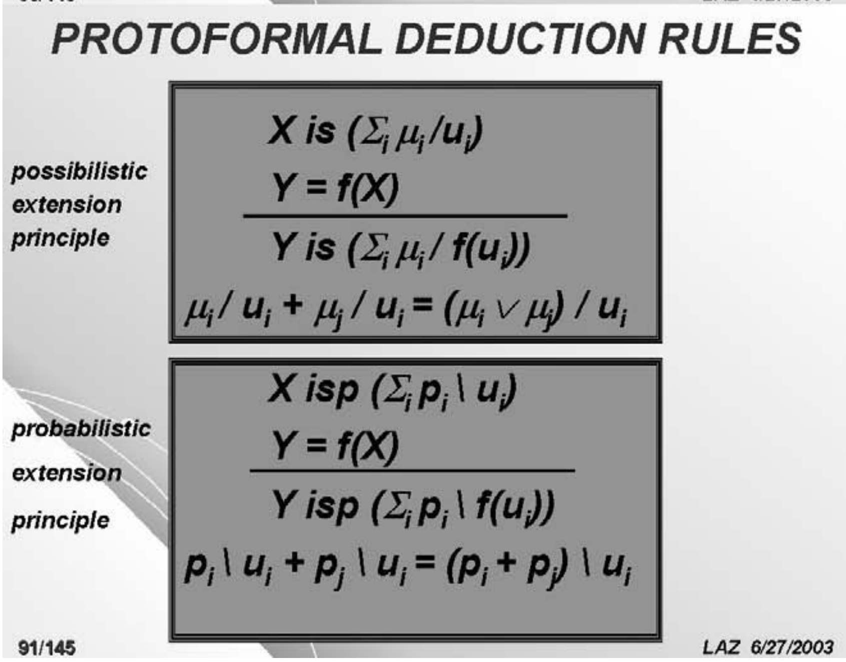
$$\frac{\text{Prob }\{X \text{ is } A\} \text{ is } P}{\text{Prob }\{f(X) \text{ is } B\} \text{ is } Q}$$

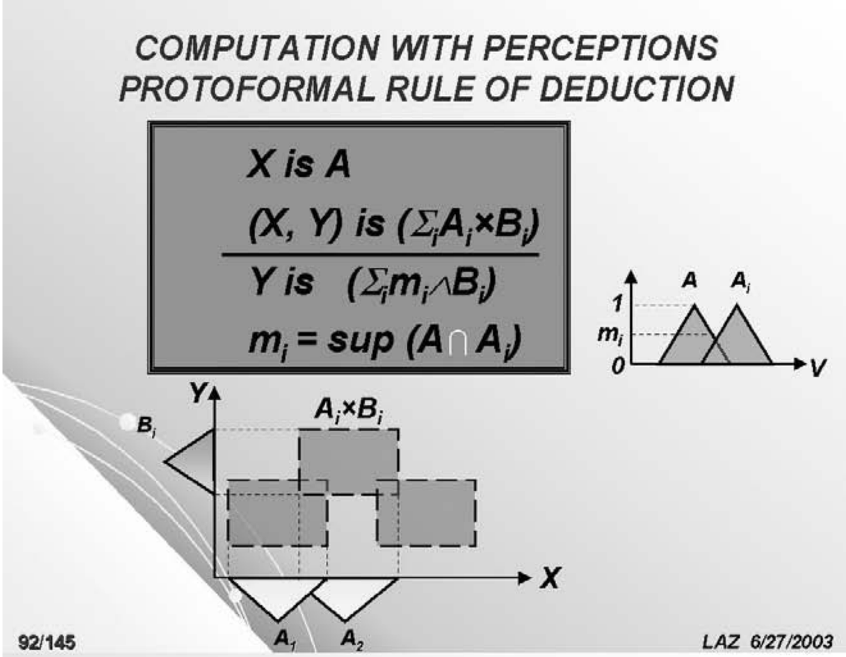
$$\frac{\begin{array}{c}X \text{ isp } P\\ (X,Y) \text{ is } R\end{array}}{Y \text{ isrs } S}$$

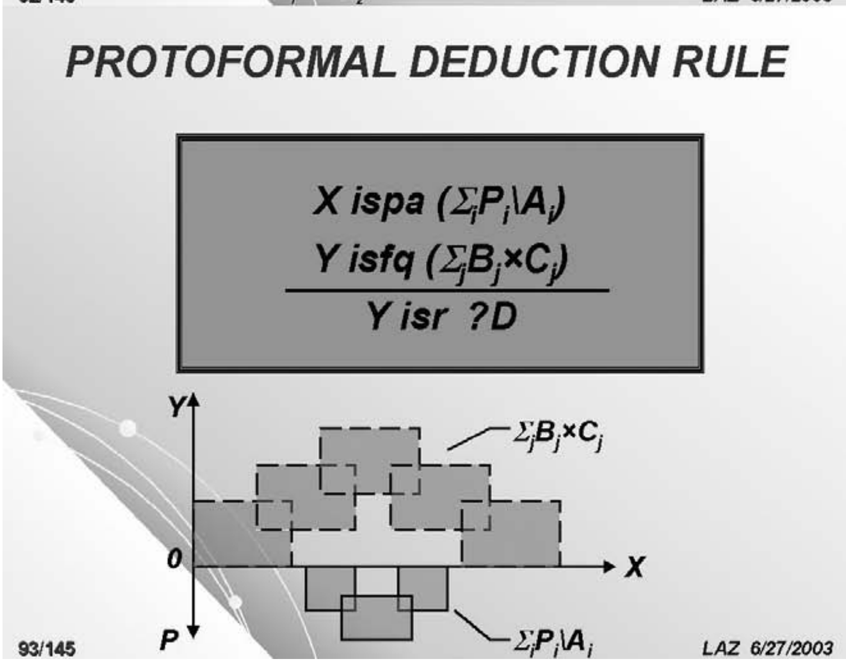
$$\frac{\begin{array}{c}X \text{ isu } A\\ Y = f(X)\end{array}}{Y \text{ isu } f(A)}$$

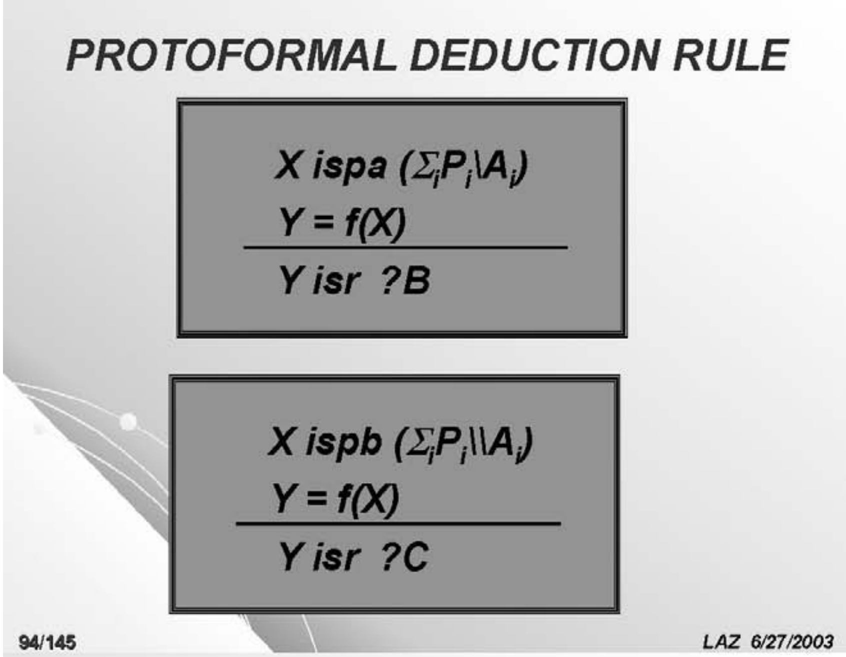
89/145 LAZ 6/27/2003

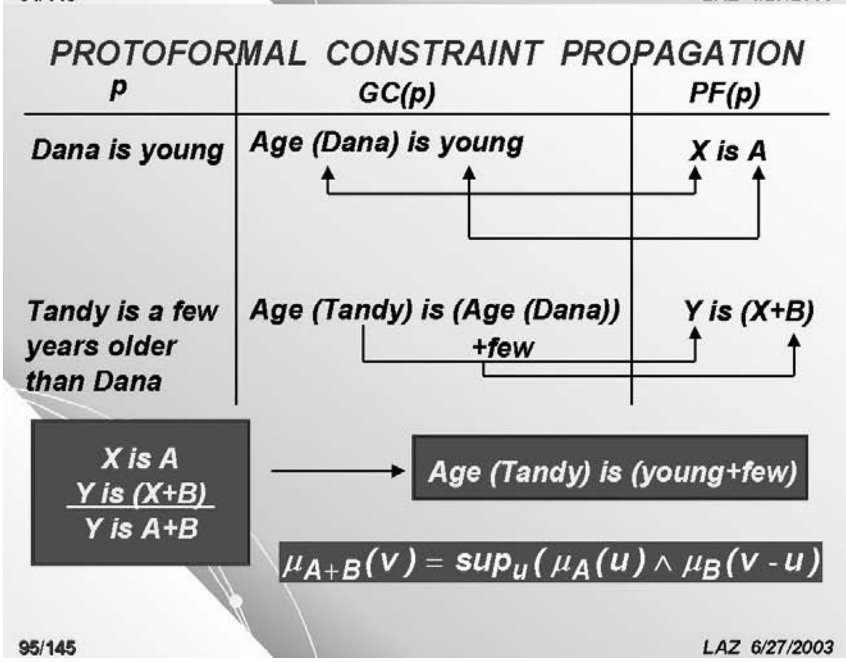

PROBABILISTIC CONSTRAINT PROPAGATION RULE
(a special version of the generalized extension principle)
$\int_U g(u)\mu_A(u)du$ is R
$\int_U g(u)\mu_B(u)du$ is ?S
$\mu_S(v) = \sup_g(\mu_R(\int_U g(u)\mu_A(u)du))$
subject to
$v = \int_U g(u)\mu_B(u)du$
$\int_U g(u)du = 1$
90/145
LAZ 6/27/2003

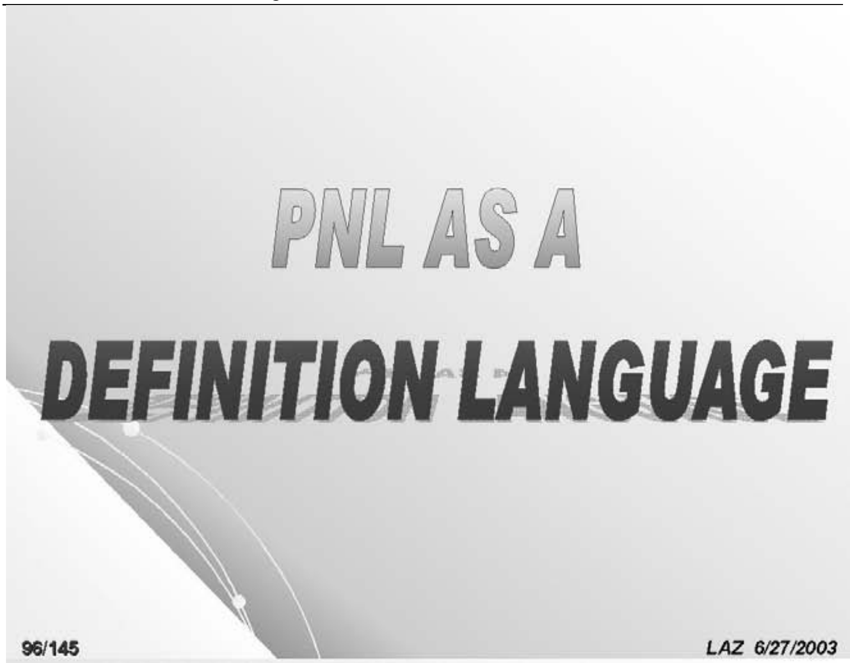

PROTOFORMAL DEDUCTION RULES
possibilistic extension principle
X is $(\Sigma_i \mu_i/u_i)$
$Y = f(X)$
Y is $(\Sigma_i \mu_i/ f(u_i))$
$\mu_i / u_i + \mu_j / u_i = (\mu_i \vee \mu_j) / u_i$
probabilistic extension principle
X isp $(\Sigma_i p_i \setminus u_i)$
$Y = f(X)$
Y isp $(\Sigma_i p_i \setminus f(u_i))$
$p_i \setminus u_i + p_j \setminus u_i = (p_i + p_j) \setminus u_i$
91/145
LAZ 6/27/2003

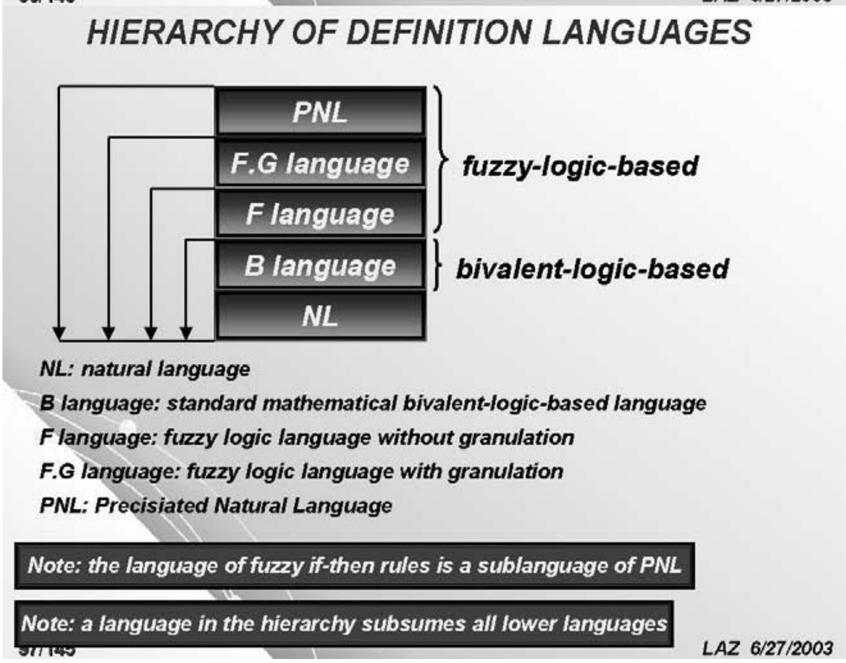

COMPUTATION WITH PERCEPTIONS
PROTOFORMAL RULE OF DEDUCTION
X is A
(X, Y) is (Σ_i A_i×B_i)
Y is (Σ_i m_i∧B_i)
m_i = sup (A∩A_i)
A
A_i
1
m_i
0
V
Y
B_i
A_i×B_i
X
A_1
A_2
92/145
LAZ 6/27/2003

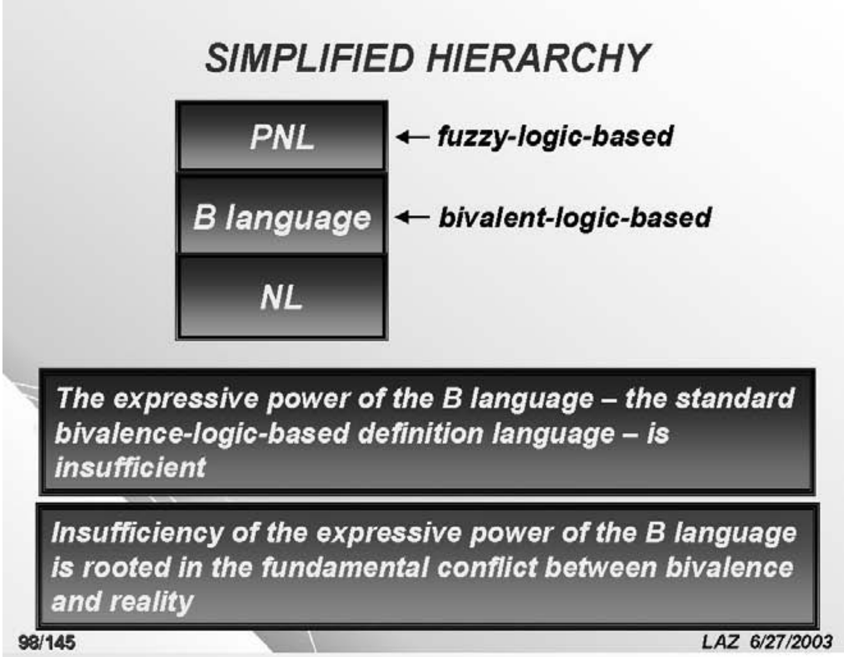

PROTOFORMAL DEDUCTION RULE
X ispa (Σ_i P_i\A_i)
Y isfq (Σ_j B_j×C_j)
Y isr ?D
Y
Σ_j B_j×C_j
0
X
P
Σ_i P_i\A_i
93/145
LAZ 6/27/2003

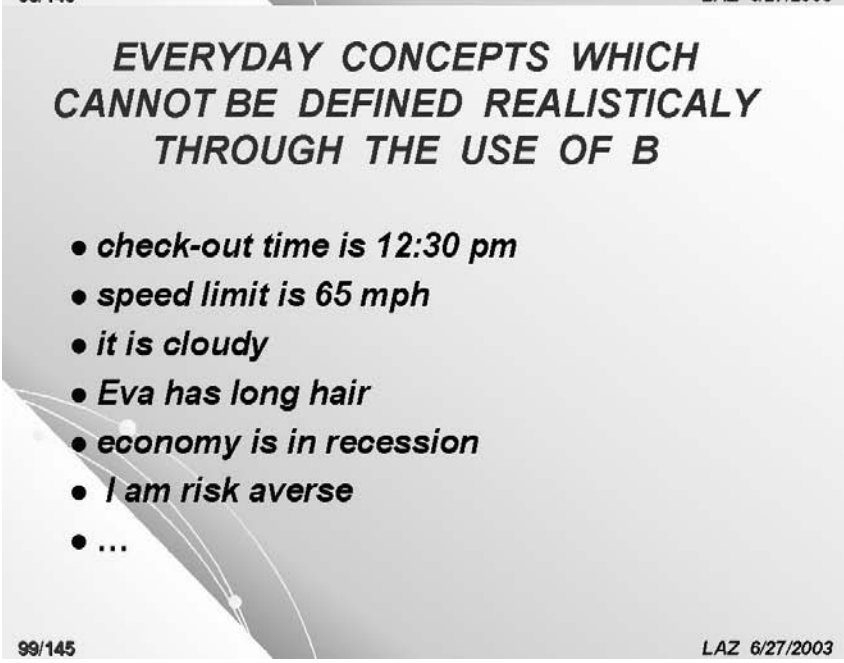

PROTOFORMAL DEDUCTION RULE
X ispa $(\Sigma_i P_i \backslash A_i)$
$Y = f(X)$
Y isr $?B$
X ispb $(\Sigma_i P_i \backslash\backslash A_i)$
$Y = f(X)$
Y isr $?C$
94/145
LAZ 6/27/2003

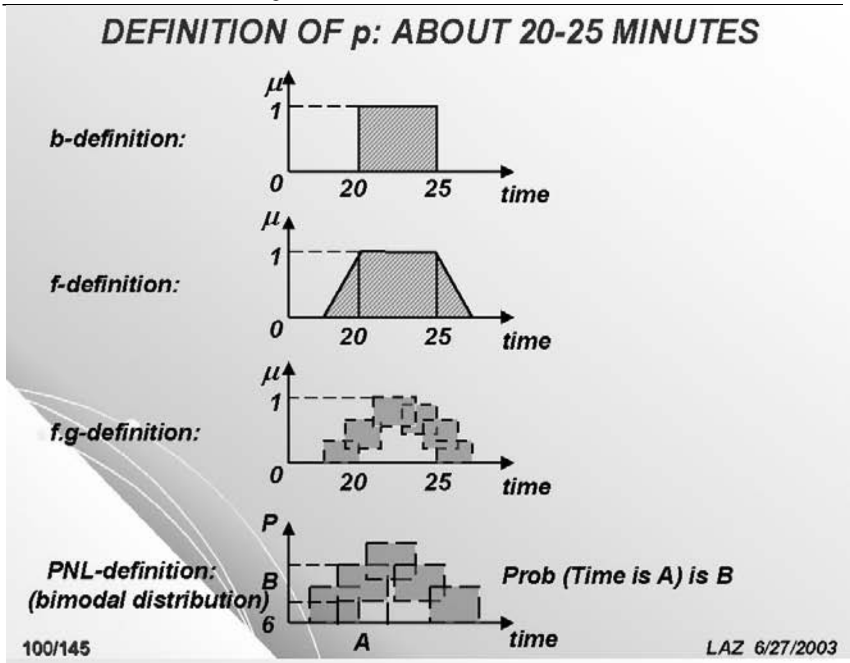

PROTOFORMAL CONSTRAINT PROPAGATION
p
GC(p)
PF(p)
Dana is young
Age (Dana) is young
X is A
Tandy is a few years older than Dana
Age (Tandy) is (Age (Dana)) +few
Y is (X+B)
X is A
Y is (X+B)
Y is A+B
Age (Tandy) is (young+few)
$\mu_{A+B}(v) = \sup_u(\mu_A(u) \wedge \mu_B(v-u))$
95/145
LAZ 6/27/2003


PNL AS A
DEFINITION LANGUAGE
96/145
LAZ 6/27/2003


HIERARCHY OF DEFINITION LANGUAGES
PNL
F.G language
F language
B language
NL
fuzzy-logic-based
bivalent-logic-based
NL: natural language
B language: standard mathematical bivalent-logic-based language
F language: fuzzy logic language without granulation
F.G language: fuzzy logic language with granulation
PNL: Precisiated Natural Language
Note: the language of fuzzy if-then rules is a sublanguage of PNL
Note: a language in the hierarchy subsumes all lower languages
LAZ 6/27/2003


SIMPLIFIED HIERARCHY
PNL
← fuzzy-logic-based
B language
← bivalent-logic-based
NL
The expressive power of the B language – the standard bivalence-logic-based definition language – is insufficient
Insufficiency of the expressive power of the B language is rooted in the fundamental conflict between bivalence and reality
98/145
LAZ 6/27/2003


EVERYDAY CONCEPTS WHICH CANNOT BE DEFINED REALISTICALY THROUGH THE USE OF B
• check-out time is 12:30 pm
• speed limit is 65 mph
• it is cloudy
• Eva has long hair
• economy is in recession
• I am risk averse
• …
99/145
LAZ 6/27/2003



INSUFFICIENCY OF THE B LANGUAGE

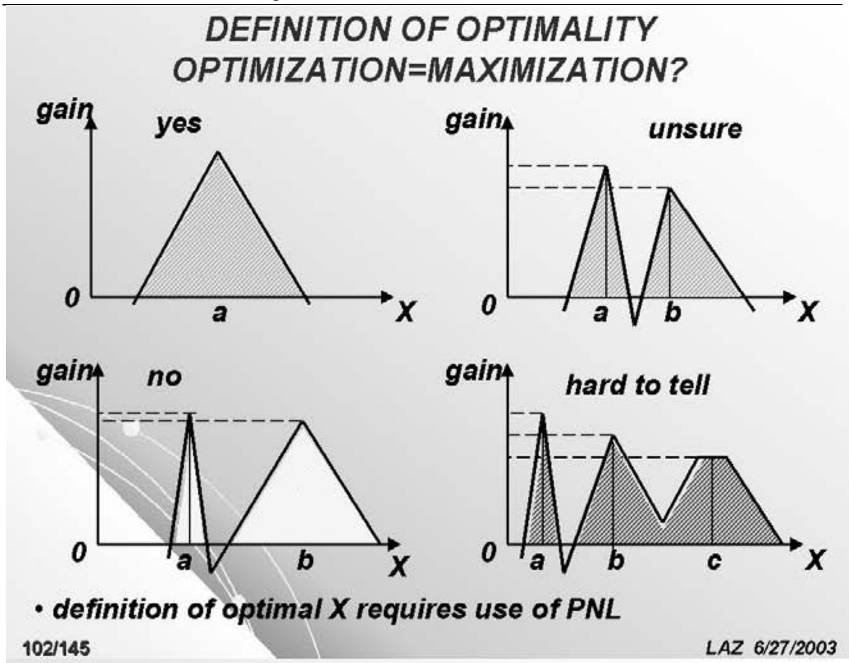
Concepts which cannot be defined

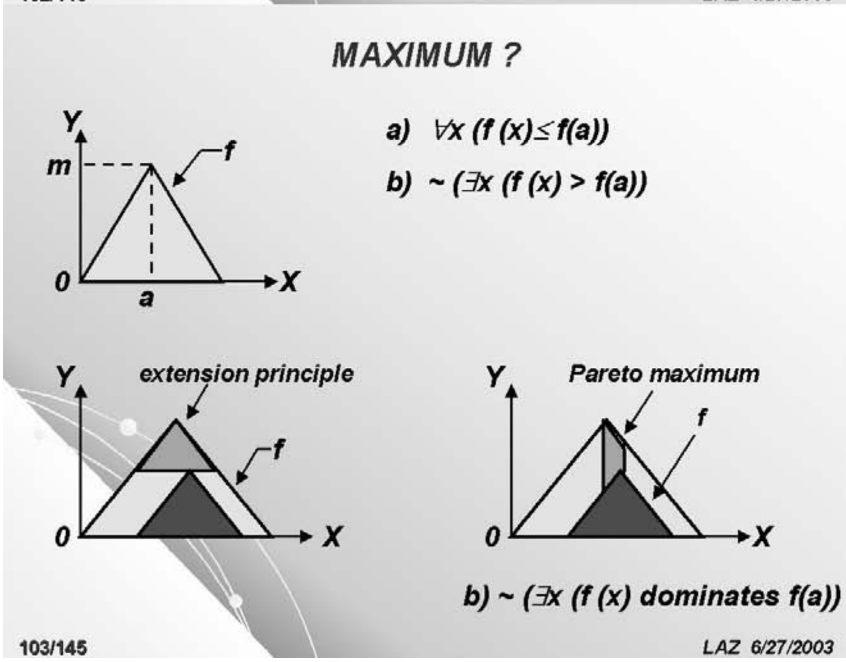
- ***causality***
- ***relevance***
- ***intelligence***

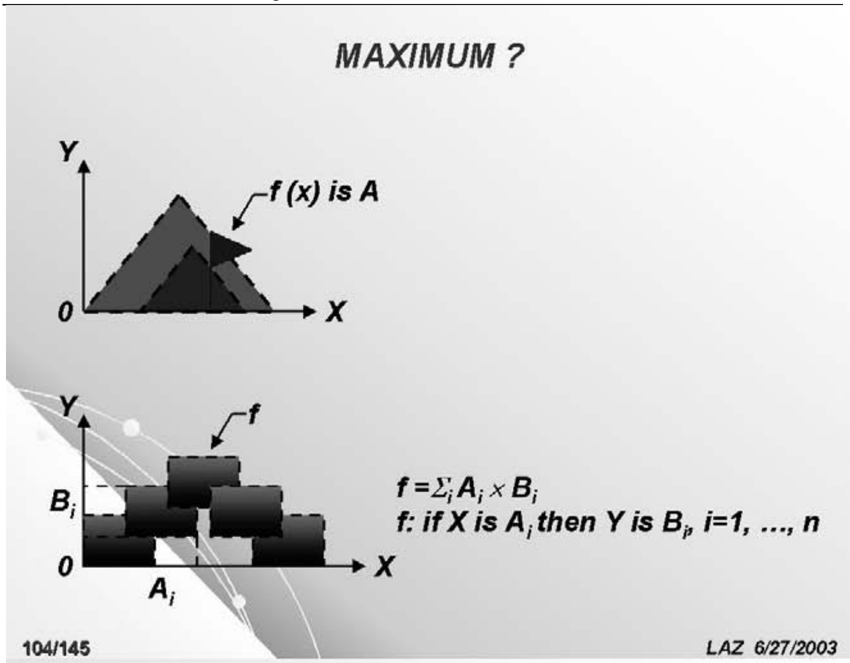
Concepts whose definitions are problematic

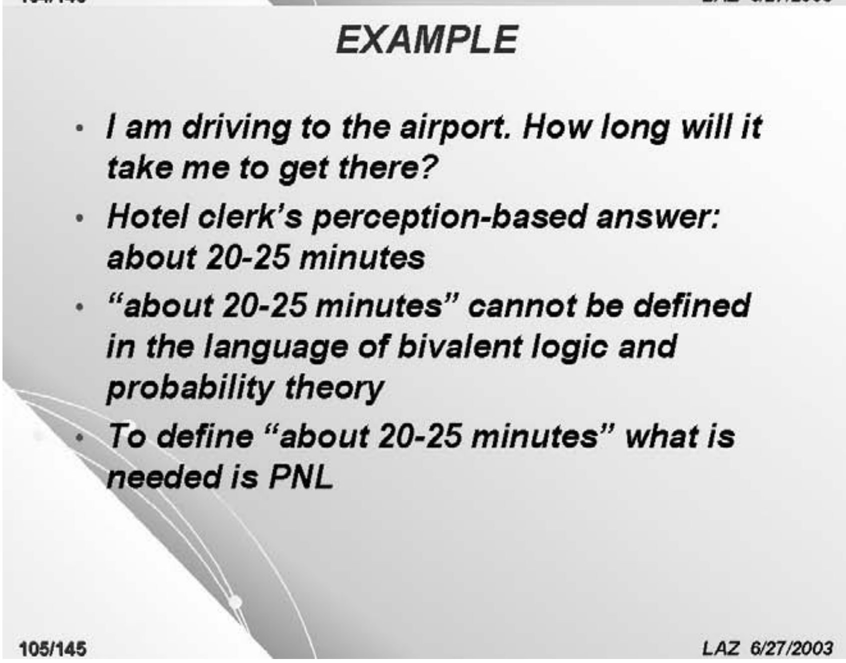
- ***stability***
- ***optimality***
- ***statistical independence***
- ***stationarity***

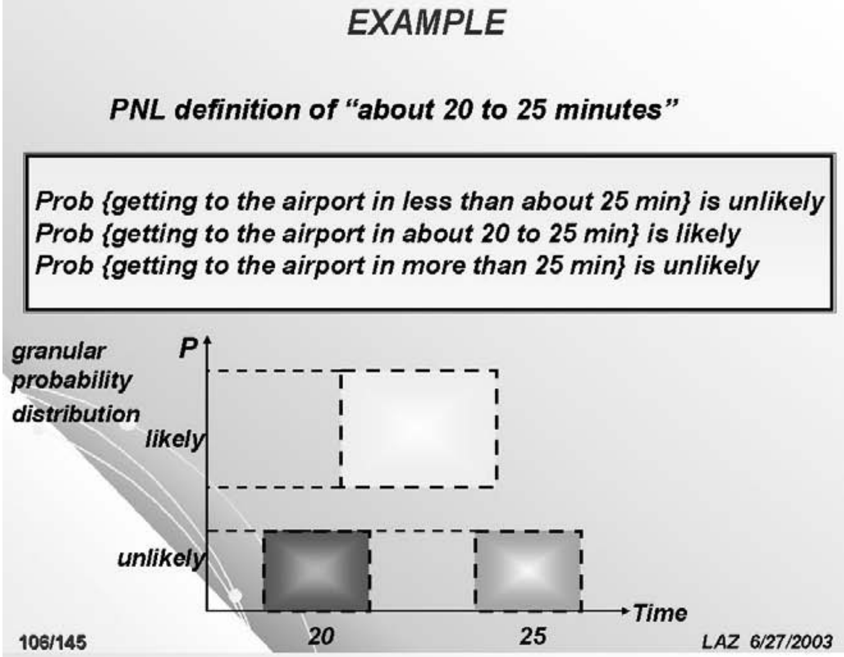
101/145 *LAZ 6/27/2003*

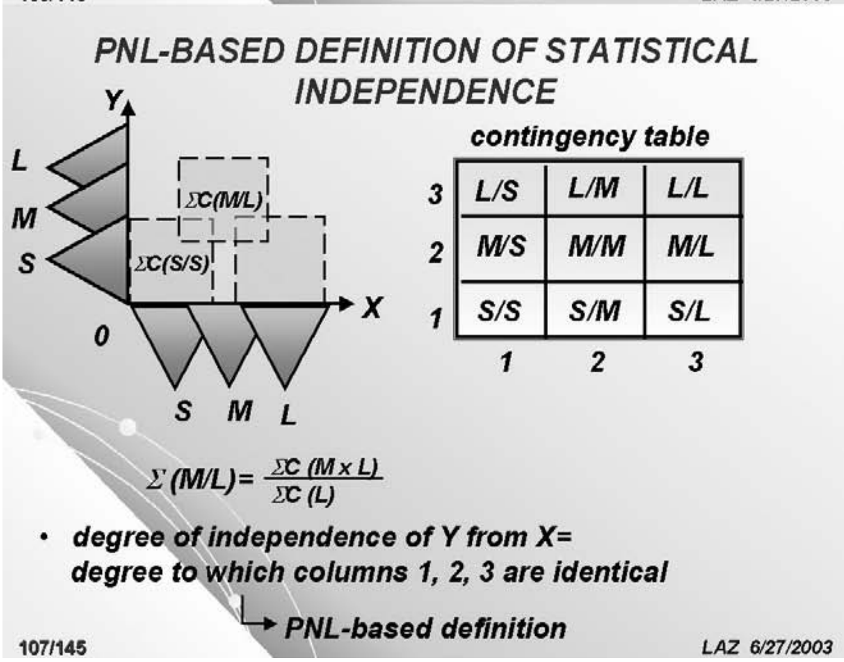

DEFINITION OF OPTIMALITY
OPTIMIZATION=MAXIMIZATION?
gain
yes
0
a
X
gain
unsure
0
a
b
X
gain
no
0
a
b
X
gain
hard to tell
0
a
b
c
X
• definition of optimal X requires use of PNL
102/145
LAZ 6/27/2003

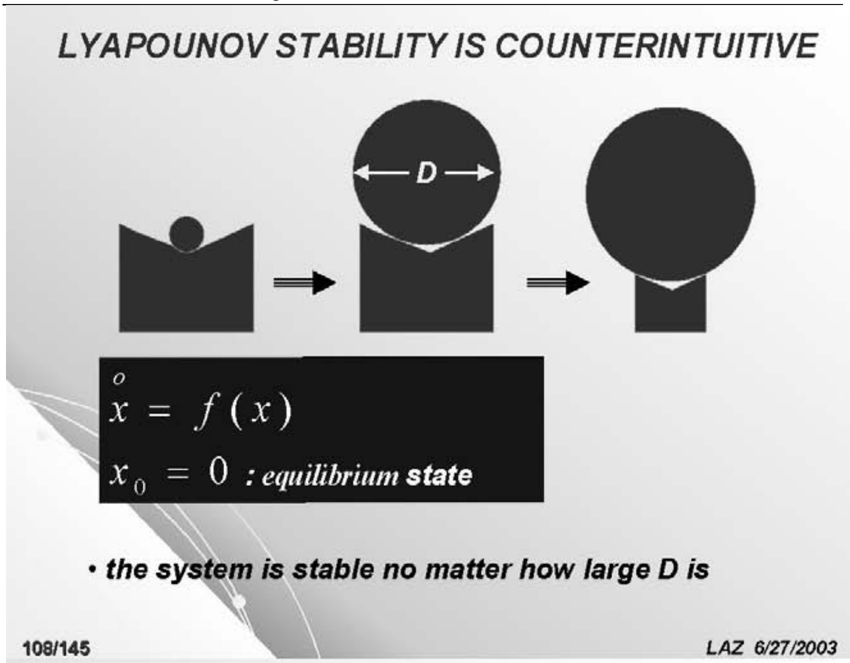

MAXIMUM ?
Y
m
f
0
a
X
a) ∀x (f (x)≤ f(a))
b) ~ (∃x (f (x) > f(a))
Y
extension principle
f
0
X
Y
Pareto maximum
f
0
X
b) ~ (∃x (f (x) dominates f(a))
103/145
LAZ 6/27/2003

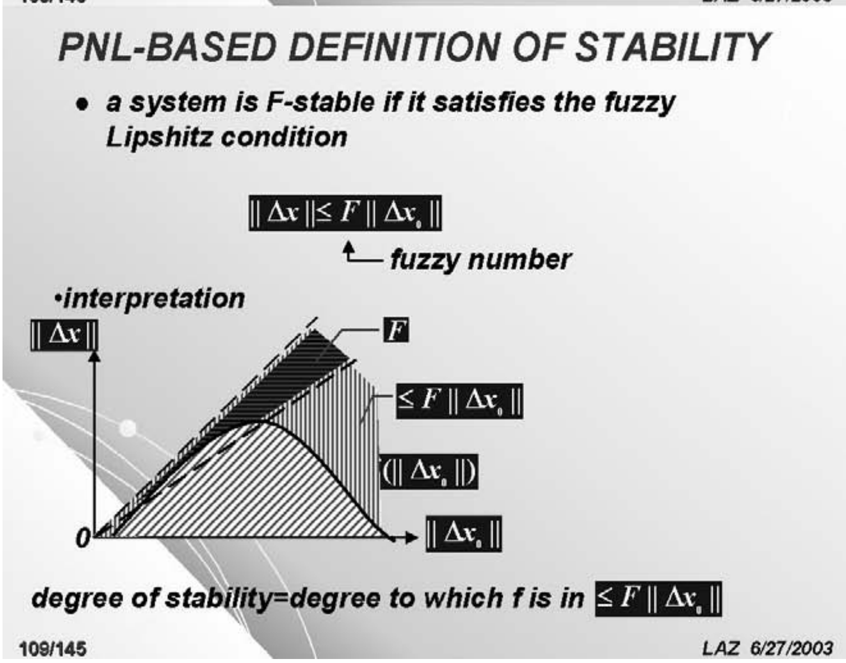

MAXIMUM ?
Y
f (x) is A
0
X
Y
f
B_i
0
A_i
X
f =Σ_i A_i × B_i
f: if X is A_i then Y is B_i, i=1, ..., n
104/145
LAZ 6/27/2003

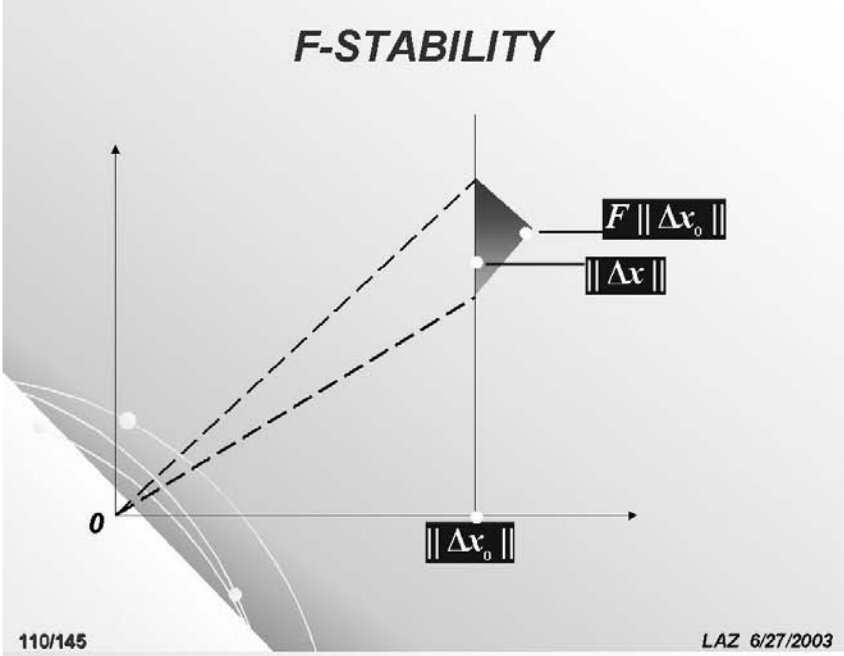

EXAMPLE
• I am driving to the airport. How long will it take me to get there?
• Hotel clerk's perception-based answer: about 20-25 minutes
• "about 20-25 minutes" cannot be defined in the language of bivalent logic and probability theory
• To define "about 20-25 minutes" what is needed is PNL
105/145
LAZ 6/27/2003


EXAMPLE
PNL definition of “about 20 to 25 minutes”
Prob {getting to the airport in less than about 25 min} is unlikely
Prob {getting to the airport in about 20 to 25 min} is likely
Prob {getting to the airport in more than 25 min} is unlikely
granular probability distribution
P
likely
unlikely
Time
20
25
106/145
LAZ 6/27/2003


PNL-BASED DEFINITION OF STATISTICAL INDEPENDENCE
Y
L
M
S
0
X
S M L
ΣC(M/L)
ΣC(S/S)
contingency table
3 L/S L/M L/L
2 M/S M/M M/L
1 S/S S/M S/L
1 2 3
Σ (M/L)= ΣC (M x L) / ΣC (L)
• degree of independence of Y from X= degree to which columns 1, 2, 3 are identical
PNL-based definition
107/145
LAZ 6/27/2003


LYAPOUNOV STABILITY IS COUNTERINTUITIVE
D
$\overset{o}{x} = f(x)$
$x_0 = 0$: equilibrium state
• the system is stable no matter how large D is
108/145
LAZ 6/27/2003


PNL-BASED DEFINITION OF STABILITY
• a system is F-stable if it satisfies the fuzzy Lipshitz condition
$\| \Delta x \| \leq F \| \Delta x_0 \|$
fuzzy number
•interpretation
$\| \Delta x \|$
F
$\leq F \| \Delta x_0 \|$
$(\| \Delta x_0 \|)$
0
$\| \Delta x_0 \|$
degree of stability=degree to which f is in $\leq F \| \Delta x_0 \|$
109/145
LAZ 6/27/2003



HIGHER-ORDER CONCEPTS

- ***What is not widely recognized is that some seemingly simple concepts, e.g., cluster and edge, are hard to define because they are higher-order concepts.***

- ***Informally, a concept is of order (level) k if its denotation is a set of order k. A set whose elements are points is of order one. A set whose elements are sets of order one is of order two, etc.***

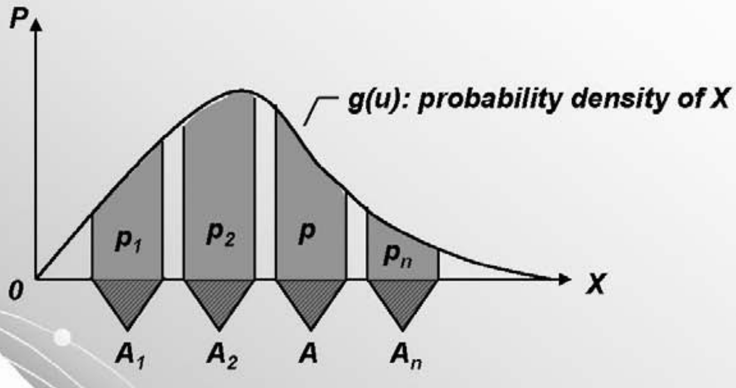
111/145 LAZ 6/27/2003

CONTINUED

- ***There are four categories of second-order concepts: (1) b.b-concepts, I.e., bivalent (crisp) concepts whose instances are bivalent sets, e.g., convex set; (2) b.f-concepts, I.e., bivalent concepts whose instances are fuzzy sets, e.g., convex fuzzy sets; (3) f.b-concepts, I.e., fuzzy concepts whose instances are bivalent sets, e.g., small squares; and (4) f.f-concepts, I.e., fuzzy concepts whose instances are fuzzy sets. The concepts of cluster and edge are examples of f.f-concepts. That is why they are hard to define.***

112/145 LAZ 6/27/2003

INTERPOLATION OF BIMODAL DISTRIBUTIONS



p_i is P_i : granular value of p_i , i=1, ..., n
(P_i, A_i) , i=1, ..., n are given
A is given
(?P, A)

113/145 LAZ 6/27/2003

INTERPOLATION MODULE AND PROBABILITY MODULE

Prob {X is A$_i$} is P$_i$, i = 1, ..., n
Prob {X is A} is Q

$$\mu_Q(v) = sup_g(\mu_{P_1}(\int_U \mu_{A_1}(u)g(u)du) \wedge \cdots \wedge$$

$$\mu_{P_n}\int_U \mu_{P_n}(\int_U \mu_{A_n}(u)g(u)du))$$

subject to

$$U = \int_U \mu_A(u)g(u)du$$

114/145 LAZ 6/27/2003

PROBABILISTIC CONSTRAINT PROPAGATION RULE
(a special version of the generalized extension principle)

$$\frac{\int_U g(u)\mu_A(u)du \quad \text{is } R}{\int_U g(u)\mu_B(u)du \quad \text{is } ?S}$$

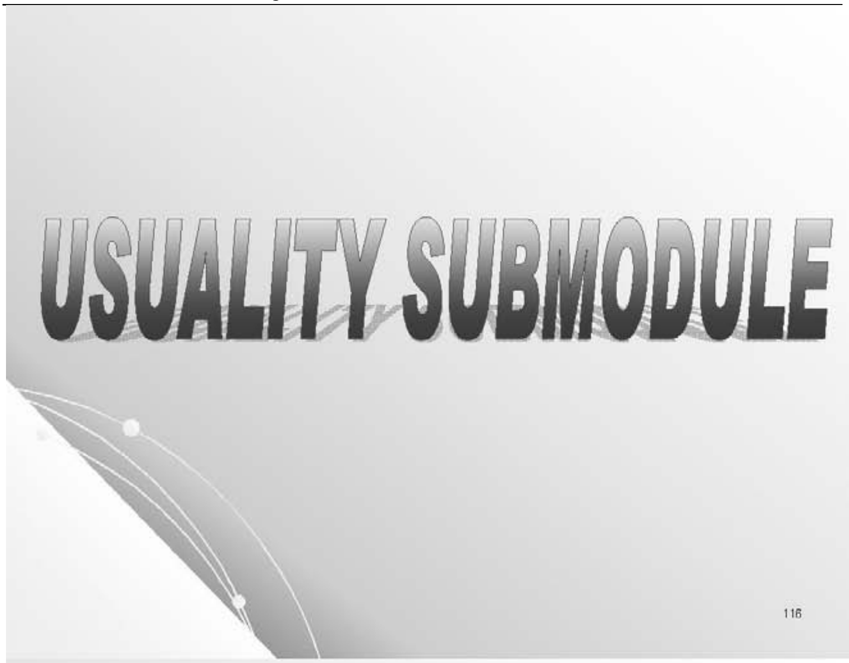
$$\mu_S(v) = \sup_g(\mu_R(\int_U g(u)\mu_A(u)du))$$

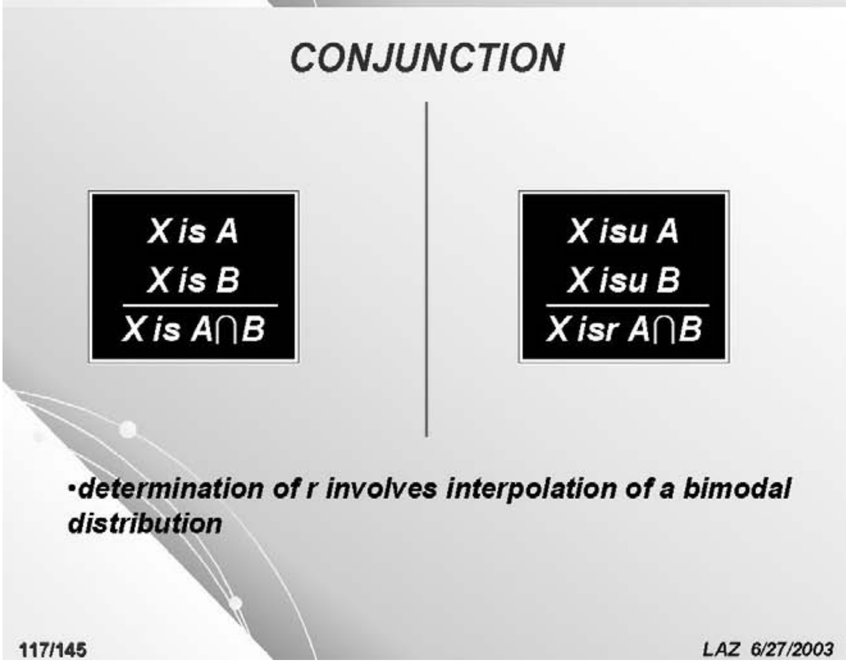
subject to

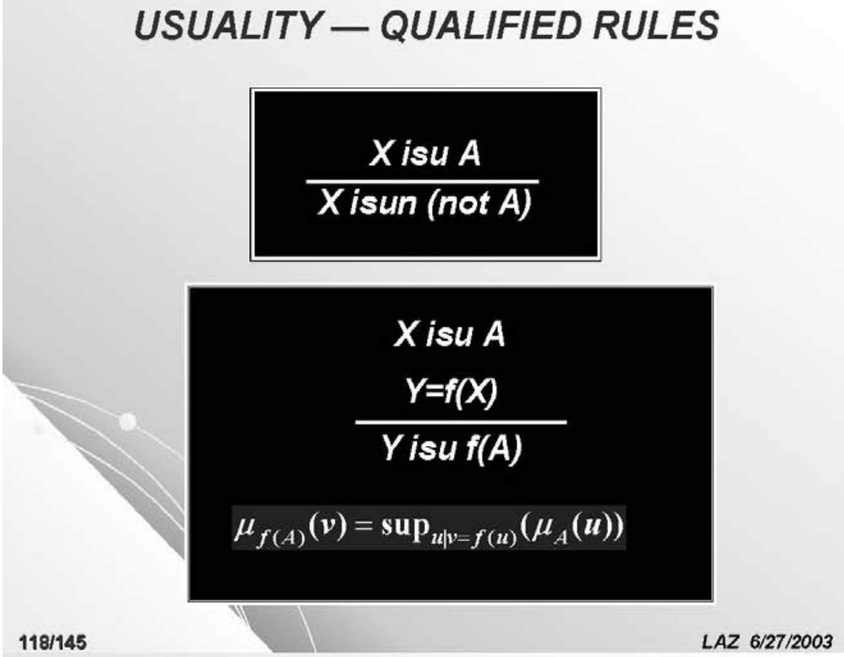
$$v = \int_U g(u)\mu_B(u)du$$

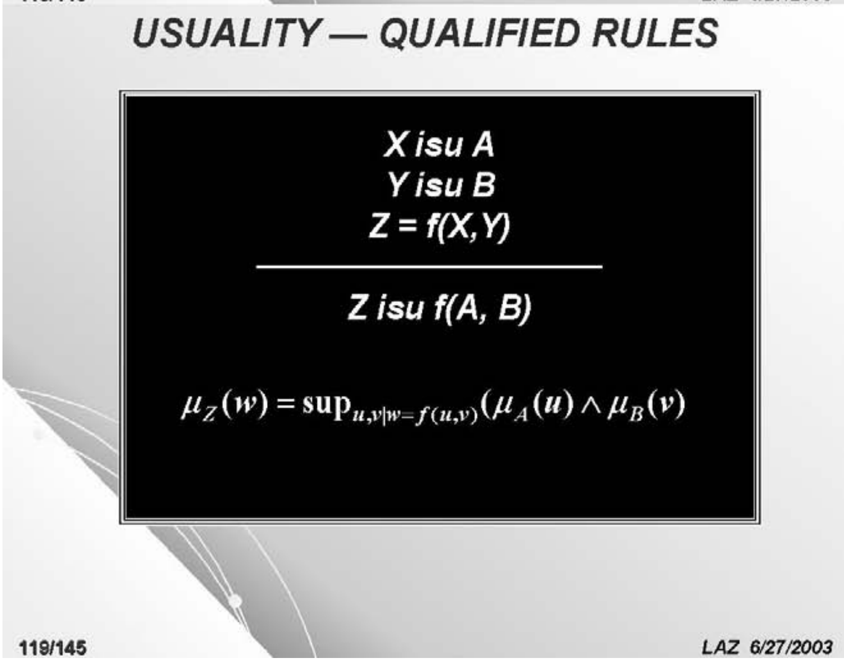
$$\int_U g(u)du = 1$$

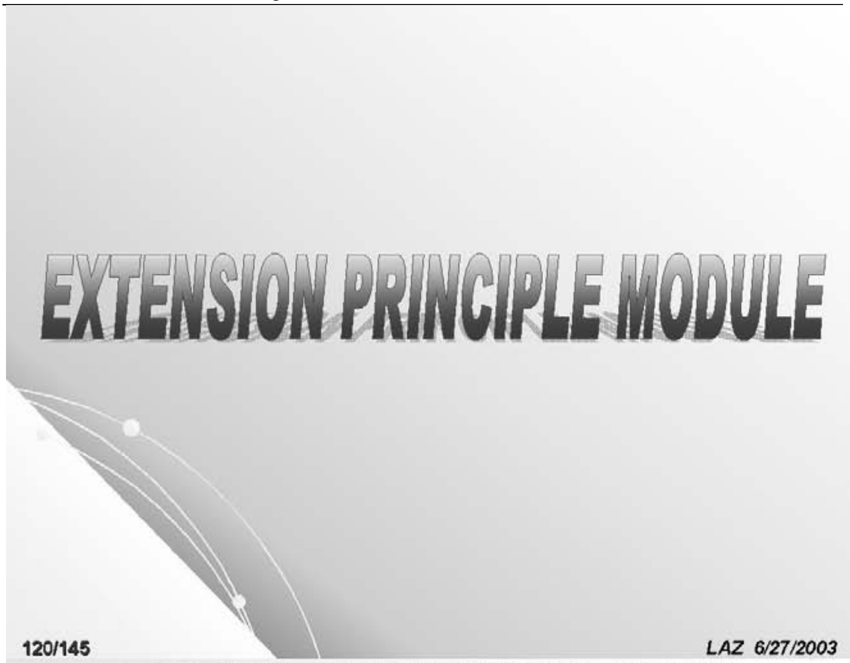
115/145 LAZ 6/27/2003

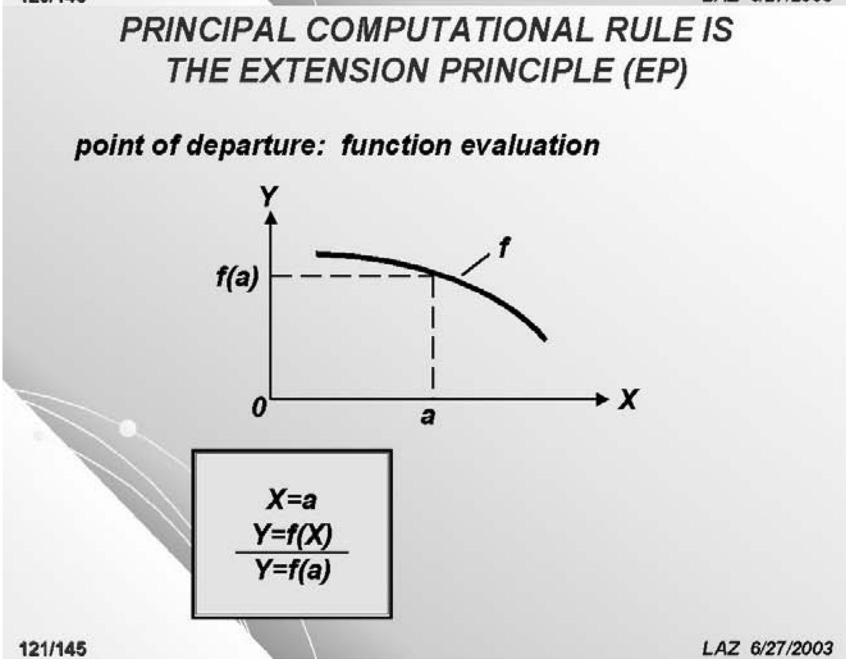

USUALITY SUBMODULE
116

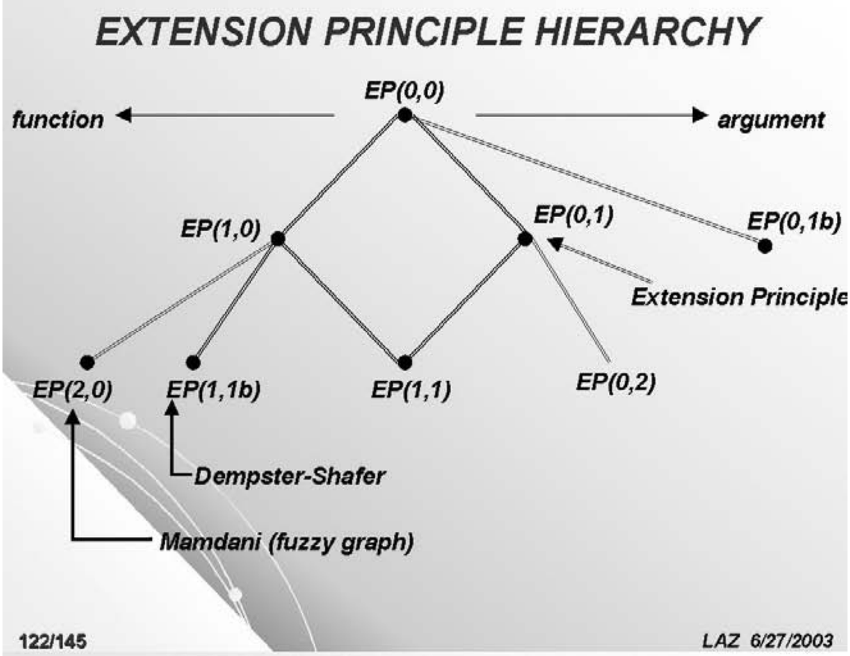

CONJUNCTION
X is A
X is B
X is A∩B
X isu A
X isu B
X isr A∩B
•determination of r involves interpolation of a bimodal distribution
117/145
LAZ 6/27/2003

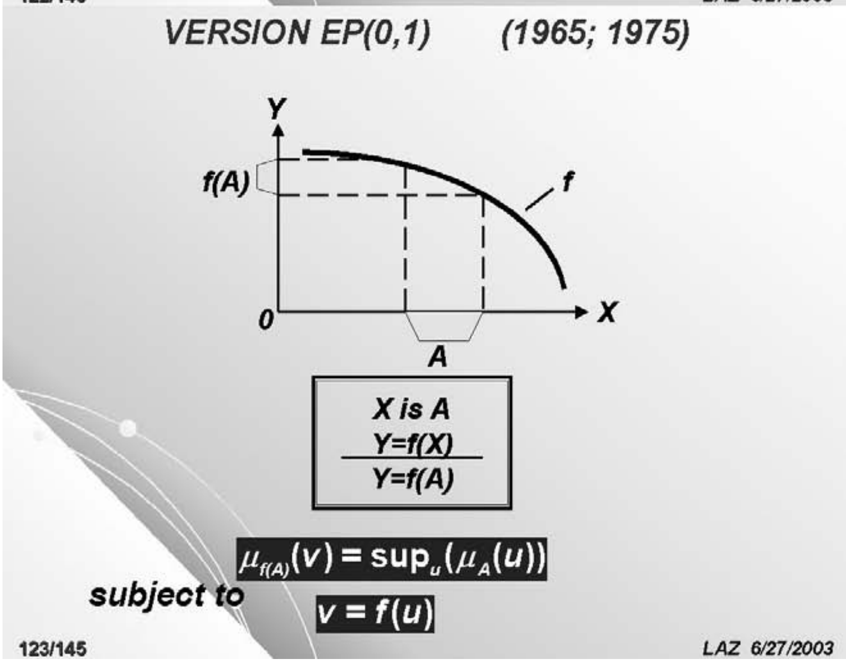

USUALITY — QUALIFIED RULES
X isu A
X isun (not A)
X isu A
Y=f(X)
Y isu f(A)
$\mu_{f(A)}(v) = \sup_{u|v=f(u)}(\mu_A(u))$
118/145
LAZ 6/27/2003

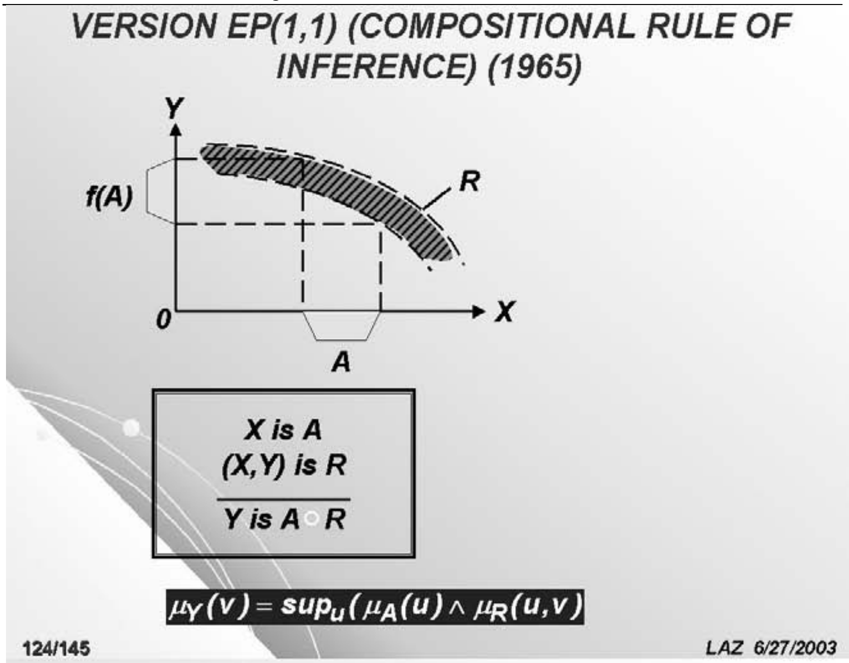

USUALITY — QUALIFIED RULES
X isu A
Y isu B
Z = f(X,Y)
Z isu f(A, B)
$\mu_Z(w) = \sup_{u,v|w=f(u,v)}(\mu_A(u) \wedge \mu_B(v)$
119/145
LAZ 6/27/2003

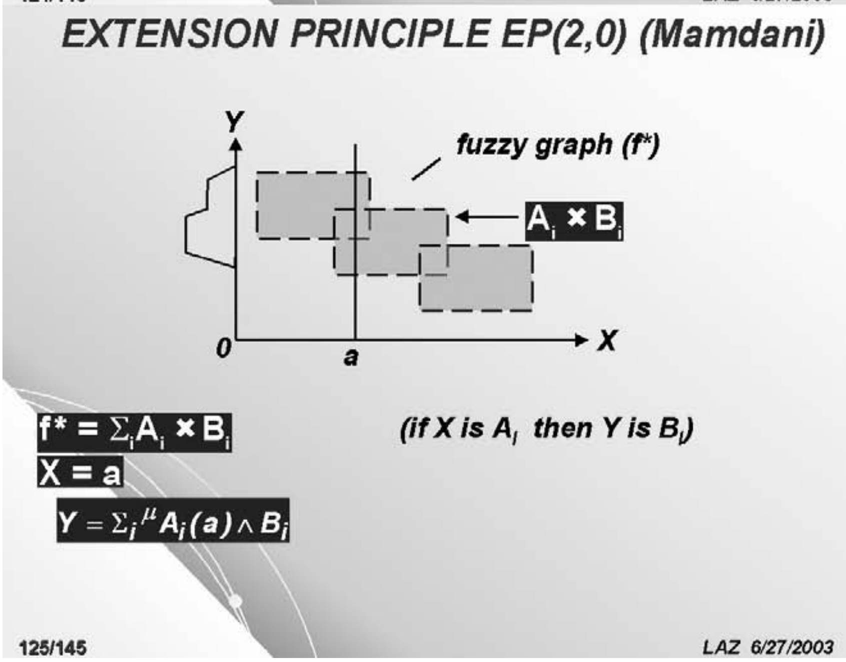

EXTENSION PRINCIPLE MODULE
120/145
LAZ 6/27/2003


PRINCIPAL COMPUTATIONAL RULE IS
THE EXTENSION PRINCIPLE (EP)
point of departure: function evaluation
Y
f
f(a)
0
a
X
X=a
Y=f(X)
Y=f(a)
121/145
LAZ 6/27/2003

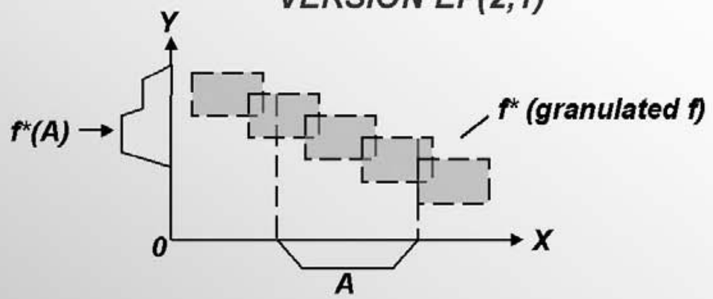

EXTENSION PRINCIPLE HIERARCHY
EP(0,0)
function
argument
EP(1,0)
EP(0,1)
EP(0,1b)
Extension Principle
EP(2,0)
EP(1,1b)
EP(1,1)
EP(0,2)
Dempster-Shafer
Mamdani (fuzzy graph)
122/145
LAZ 6/27/2003


VERSION EP(0,1) (1965; 1975)
Y
f(A)
f
0
X
A
X is A
Y=f(X)
Y=f(A)
$\mu_{f(A)}(v) = \sup_u(\mu_A(u))$
subject to
$v = f(u)$
123/145
LAZ 6/27/2003


VERSION EP(1,1) (COMPOSITIONAL RULE OF INFERENCE) (1965)
Y
f(A)
R
0
X
A
X is A
(X,Y) is R
Y is A ∘ R
$\mu_Y(v) = sup_u(\mu_A(u) \wedge \mu_R(u,v))$
124/145
LAZ 6/27/2003


EXTENSION PRINCIPLE EP(2,0) (Mamdani)
Y
fuzzy graph (f*)
$A_i \times B_i$
0
a
X
$f^* = \Sigma_i A_i \times B_i$
(if X is A_i then Y is B_i)
X = a
$Y = \Sigma_i {}^{\mu}A_i(a) \wedge B_i$
125/145
LAZ 6/27/2003

VERSION EP(2,1)



$$\frac{\begin{array}{c}X \text{ is } A \\ (X, Y) \text{ is } R\end{array}}{Y \text{ is } \Sigma_i m_i \wedge B_i}$$

$$R = \Sigma_i A_i \times B_i$$

$m_i = sup_u\,(\mu_A(u) \wedge \mu_{Ai}(u))$: matching coefficient

126/145 LAZ 6/27/2003

VERSION EP(1,1b) (DEMPSTER-SHAFER)

$$\frac{\begin{array}{l}X \text{ isp } (p_1\backslash u_1 + \ldots + p_u\backslash u_n) \\ (X,Y) \text{ is } R\end{array}}{Y \text{ isp } (p_1\backslash R(u_1) + \ldots + p_n\backslash R(u_n))}$$

Y is a fuzzy-set-valued random variable

$$\mu_{R(ui)}(v) = \mu_R(u_i, v)$$

127/145 LAZ 6/27/2003

VERSION GEP(0,0)

$$\frac{f(X) \text{ is } A}{g(X) \text{ is } g(f^{-1}(A))}$$

$$\mu_{g(f^{-1}(A))}(v) = \sup_u(\mu_A(f(u)))$$

subject to

$$v = g(u)$$

128/145 LAZ 6/27/2003

GENERALIZED EXTENSION PRINCIPLE

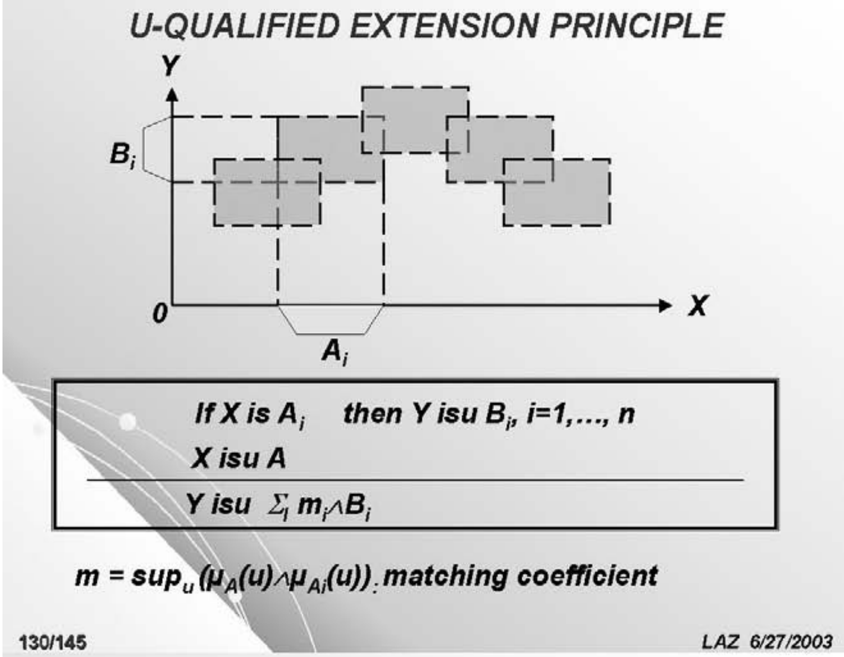
$$\frac{\begin{array}{l} f(X) \text{ is } A \\ g(Y) \text{ is } B \\ Z=h(X,Y) \end{array}}{Z \text{ is } h\ (f^{-1}(A),\ g^{-1}\ (B))}$$

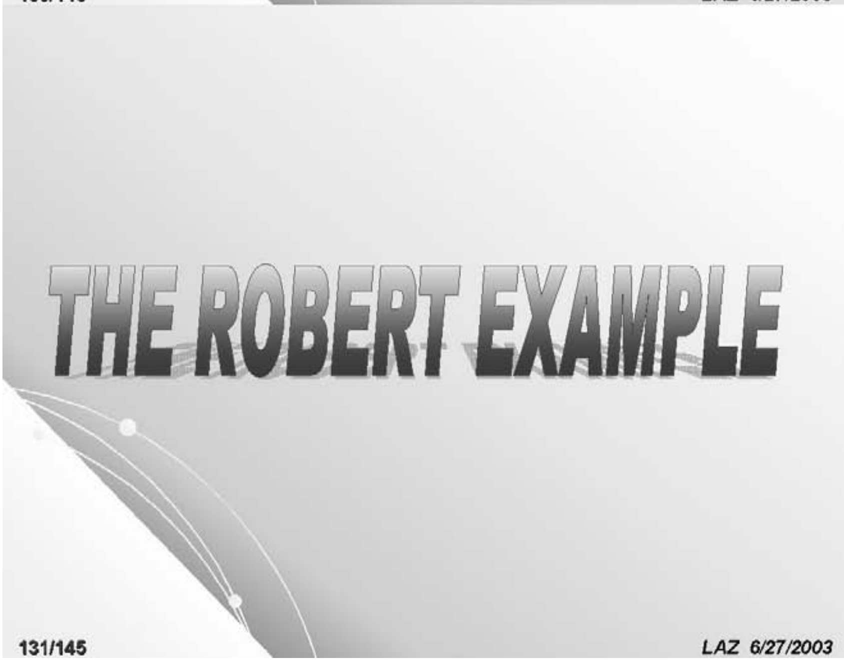
$$\mu_Z(w) = \sup_{u,v}\ (\mu_A(f(u)) \wedge \mu_B(g(u)))$$

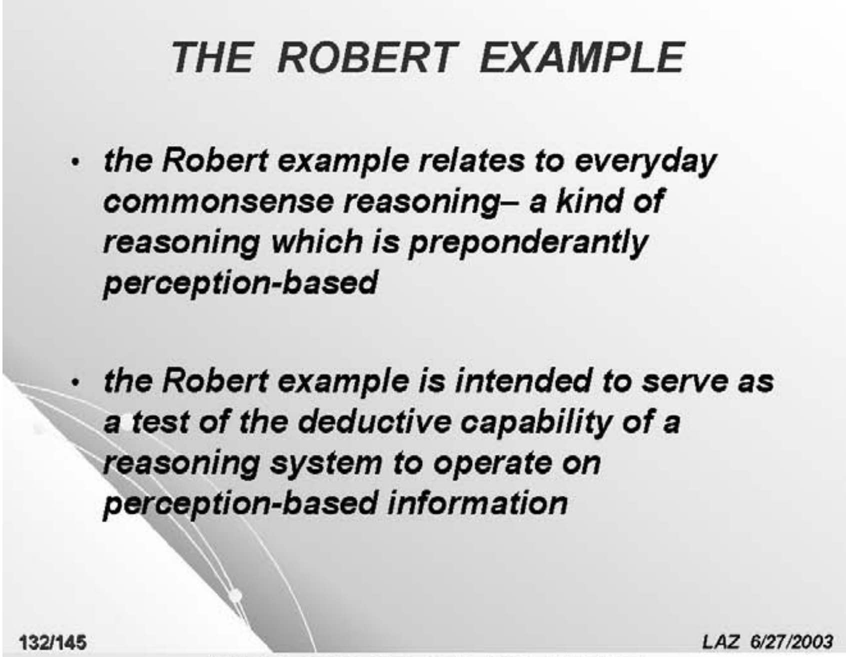
subject to

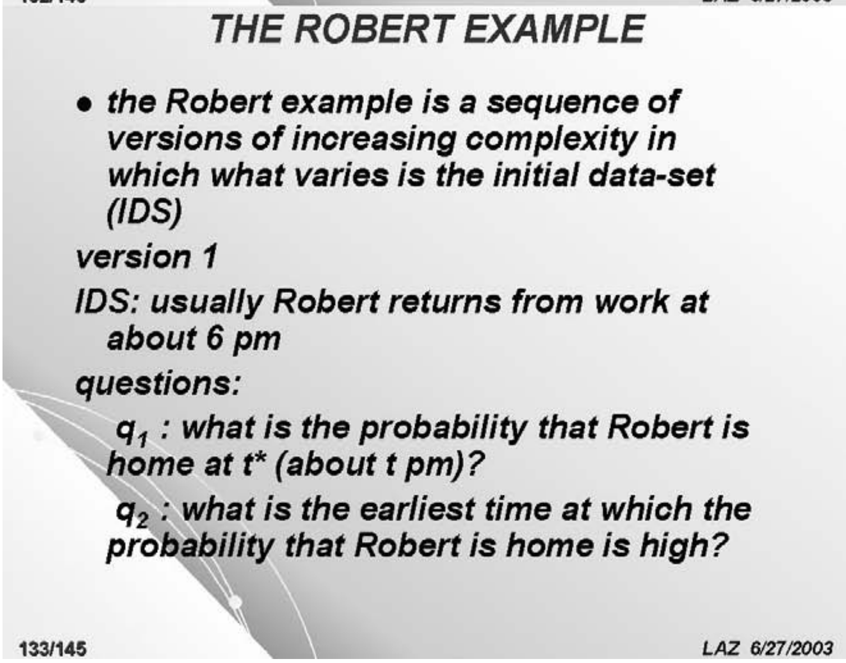
$$w = h(u,v)$$

129/145 LAZ 6/27/2003

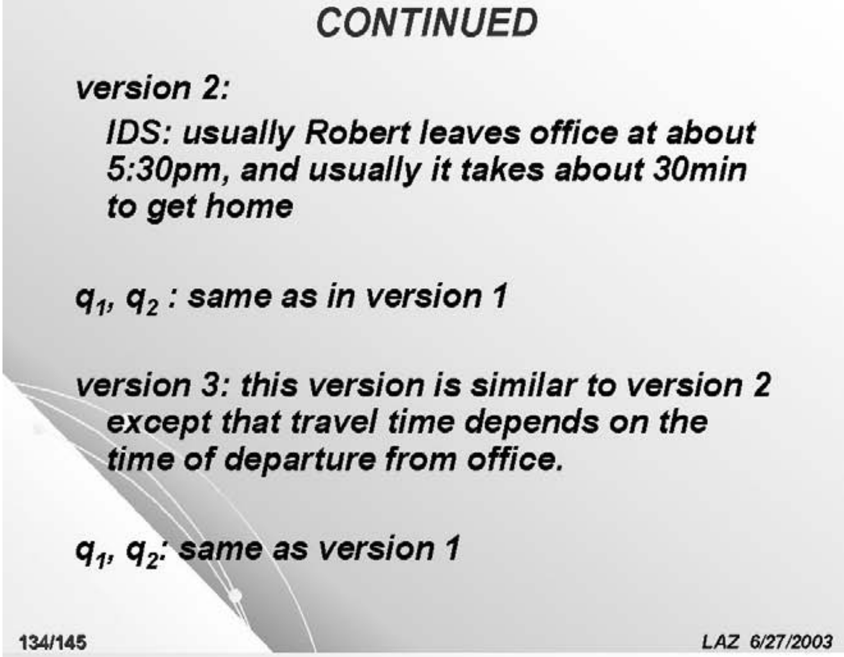

U-QUALIFIED EXTENSION PRINCIPLE
Y
B_i
0
X
A_i
If X is A_i then Y isu B_i, i=1,..., n
X isu A
Y isu $\Sigma_i\, m_i \wedge B_i$
$m = sup_u (\mu_A(u) \wedge \mu_{Ai}(u))$: matching coefficient
130/145
LAZ 6/27/2003

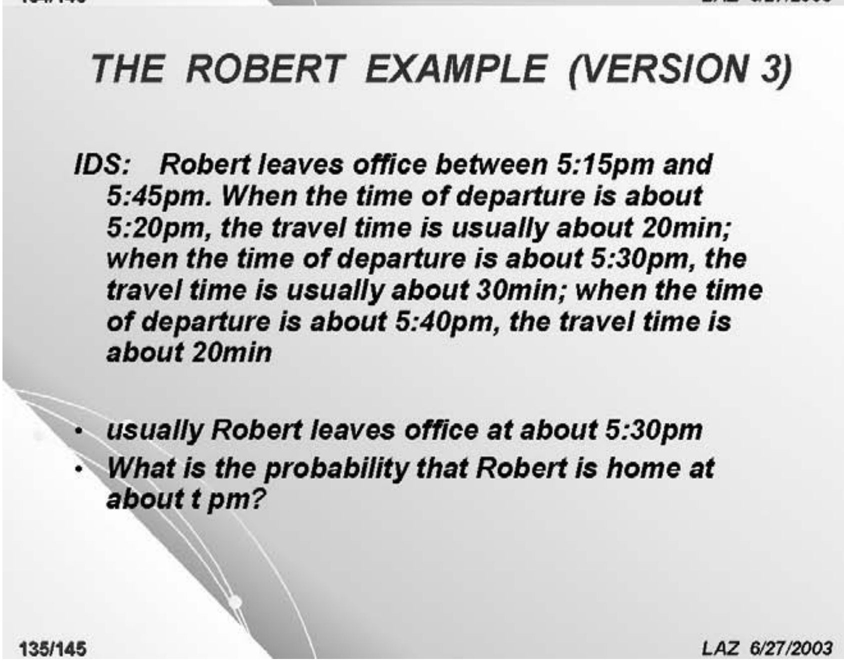

THE ROBERT EXAMPLE
131/145
LAZ 6/27/2003


THE ROBERT EXAMPLE
• the Robert example relates to everyday commonsense reasoning– a kind of reasoning which is preponderantly perception-based
• the Robert example is intended to serve as a test of the deductive capability of a reasoning system to operate on perception-based information
132/145
LAZ 6/27/2003


THE ROBERT EXAMPLE
• the Robert example is a sequence of versions of increasing complexity in which what varies is the initial data-set (IDS)
version 1
IDS: usually Robert returns from work at about 6 pm
questions:
q_1 : what is the probability that Robert is home at t* (about t pm)?
q_2 : what is the earliest time at which the probability that Robert is home is high?
133/145
LAZ 6/27/2003

Scientific and Technical Bulletin Series: Electrotechnics, Electronics, Automatic Control and Computer Science, Vol. 2, No. 1, 2005, ISSN 1584-9198

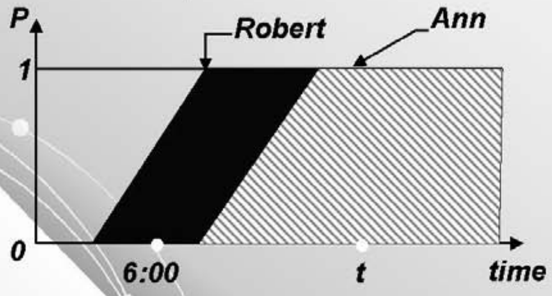




THE ROBERT EXAMPLE

Version 4

- ***Usually Robert returns from work at about 6 pm***
  Usually Ann returns from work about half-an-hour later
  What is the probability that both Robert and Ann are home at about t pm?



136/145 LAZ 6/27/2003

THE ROBERT EXAMPLE

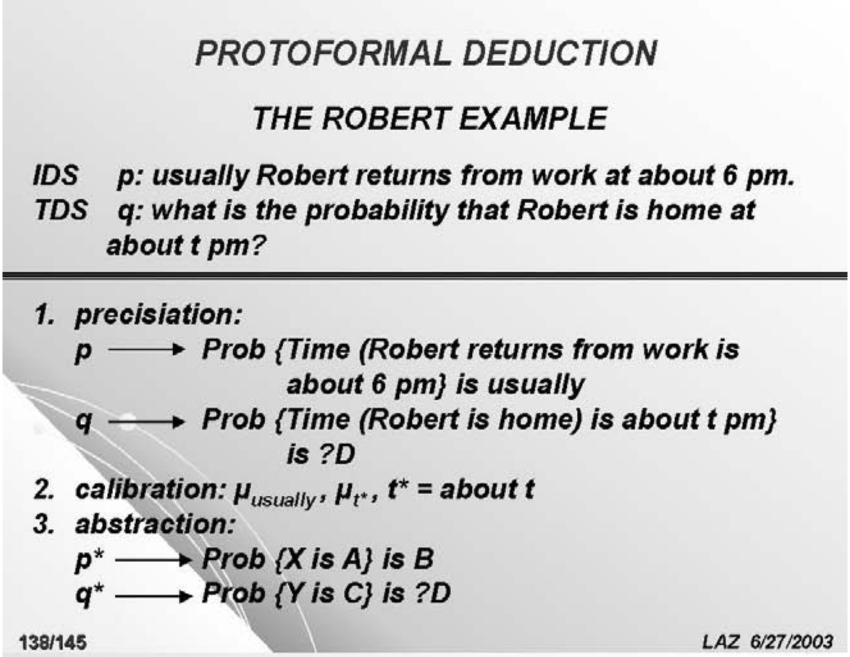
Version 1.

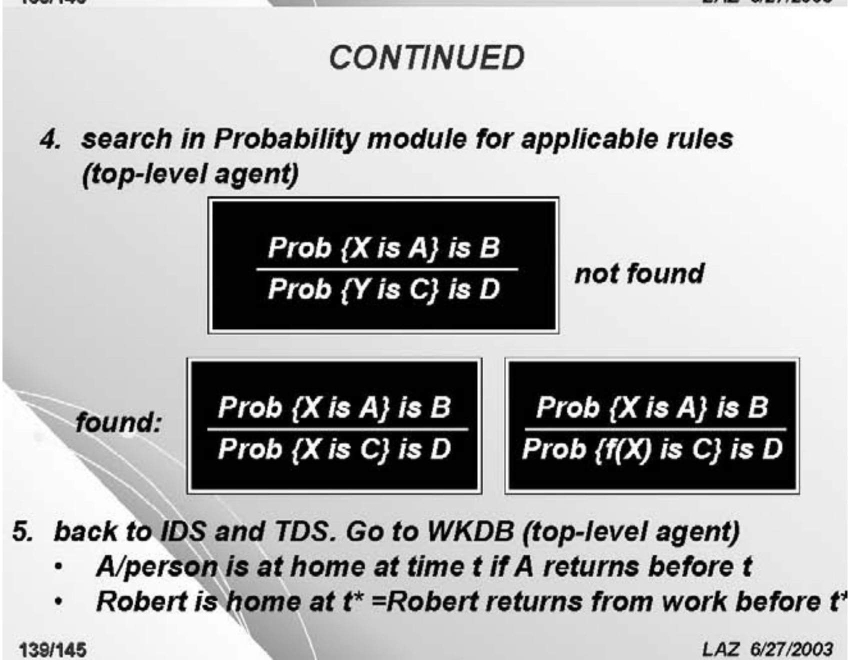
My perception is that Robert usually returns from work at about 6:00pm

q_1 : What is the probability that Robert is home at about t pm?

q_2 : What is the earliest time at which the probability that Robert is home is high?

137/145 LAZ 6/27/2003

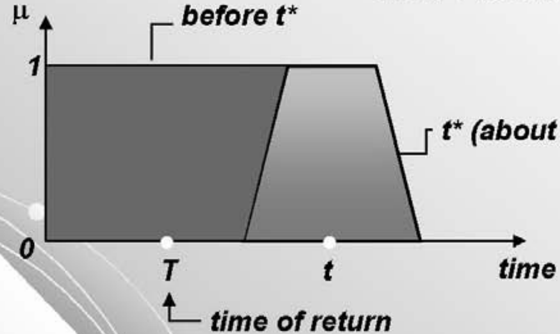

PROTOFORMAL DEDUCTION
THE ROBERT EXAMPLE
IDS p: usually Robert returns from work at about 6 pm.
TDS q: what is the probability that Robert is home at about t pm?
1. precisiation:
p → Prob {Time (Robert returns from work is about 6 pm} is usually
q → Prob {Time (Robert is home) is about t pm} is ?D
2. calibration: $\mu_{usually}$, μ_{t^*}, t* = about t
3. abstraction:
p* → Prob {X is A} is B
q* → Prob {Y is C} is ?D
138/145
LAZ 6/27/2003


CONTINUED
4. search in Probability module for applicable rules (top-level agent)
$\frac{\text{Prob \{X is A\} is B}}{\text{Prob \{Y is C\} is D}}$ not found
found: $\frac{\text{Prob \{X is A\} is B}}{\text{Prob \{X is C\} is D}}$ $\frac{\text{Prob \{X is A\} is B}}{\text{Prob \{f(X) is C\} is D}}$
5. back to IDS and TDS. Go to WKDB (top-level agent)
• A/person is at home at time t if A returns before t
• Robert is home at t* =Robert returns from work before t*
139/145
LAZ 6/27/2003

THE ROBERT EXAMPLE

event equivalence

Robert is home at about t pm= Robert returns from work before about t pm



Before about t pm= $\leq_{\circ}$ ***about t pm***

140/145 LAZ 6/27/2003

CONTINUED

6. back to Probability module

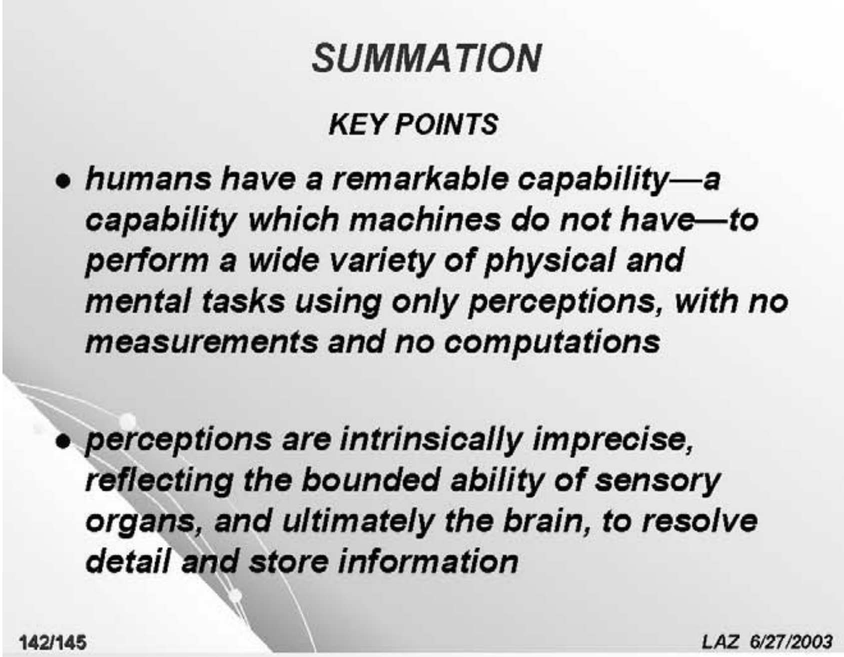
$$\frac{\text{Prob }\{X \text{ is } A\} \text{ is } B}{\text{Prob }\{X \text{ is } C\} \text{ is } D}$$

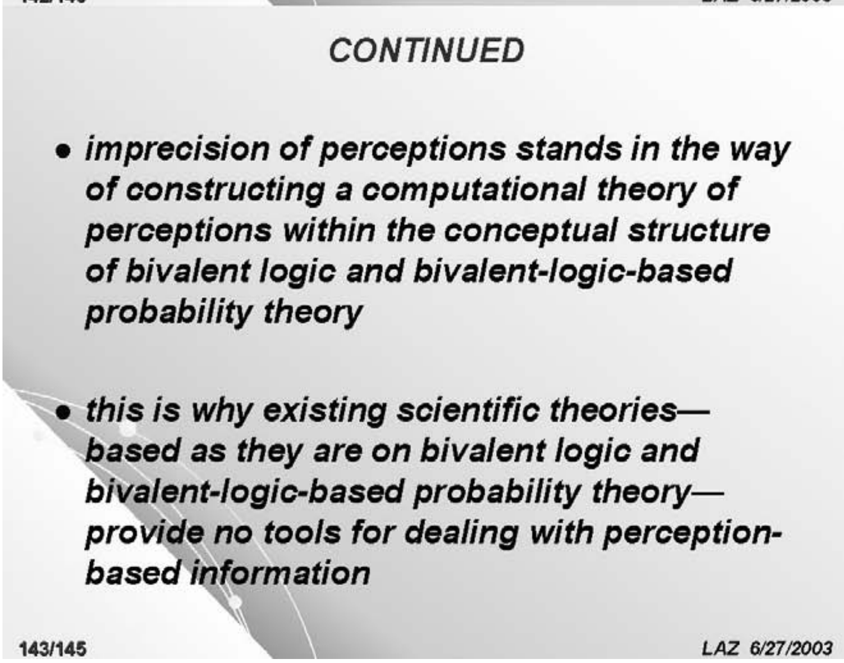
$$\mu_D(v) = sup_g(\mu_B(\int_U \mu_A(u)g(u)du))$$

$$v = \int_U \mu_c(u)g(u)du$$

7. Instantiation : ***D =Prob {Robert is home at about 6:15}***
X =Time (Robert returns from work)
A = 6*
B =usually
C = $\leq$***6:15****

141/145 LAZ 6/27/2003


SUMMATION
KEY POINTS
• humans have a remarkable capability—a capability which machines do not have—to perform a wide variety of physical and mental tasks using only perceptions, with no measurements and no computations
• perceptions are intrinsically imprecise, reflecting the bounded ability of sensory organs, and ultimately the brain, to resolve detail and store information
142/145
LAZ 6/27/2003


CONTINUED
• imprecision of perceptions stands in the way of constructing a computational theory of perceptions within the conceptual structure of bivalent logic and bivalent-logic-based probability theory
• this is why existing scientific theories—based as they are on bivalent logic and bivalent-logic-based probability theory—provide no tools for dealing with perception-based information
143/145
LAZ 6/27/2003

CONTINUED

- *in computing with words and perceptions (CWP), the objects of computation are propositions drawn from a natural language and, in particular, propositions which are descriptors of perceptions*

- *computing with words and perceptions is a methodology which may be viewed as (a) a new direction for dealing with imprecision, uncertainty and partial truth; and (b) as a basis for the analysis and design of systems which are capable of operating on perception-based information*

144/145 LAZ 6/27/2003

STATISTICS

Count of papers containing the word "fuzzy" in the title, as cited in INSPEC and MATH.SCI.NET databases. (data for 2002 are not complete)

Compiled by Camille Wanat, Head, Engineering Library, UC Berkeley, April 17, 2003

INSPEC/fuzzy		*Math.Sci.Net/fuzzy*
1970-1979	*569*	*443*
1980-1989	*2,404*	*2,466*
1990-1999	*23,207*	*5,472*
2000-present	*8,745*	*2,319*
1970-present	*34,925*	*10,700*

145/145 LAZ 6/27/2003

Marius M. BĂLAŞ

„Aurel Vlaicu" University of Arad, Engineering Faculty
Bd. Revoluției nr. 77, 310130, Arad, Romania,
E-mail: *marius.balas@ieee.org*

THE IDENTIFICATION OF THE OPERATING REGIMES OF THE CONTROLLERS BY THE HELP OF THE PHASE TRAJECTORY

Note: *The paper was presented to the International Summer School „SOFT COMPUTING" Arad-Moneasa 16.08 – 22.08.2004*

Abstract

The paper presents an on-line identification method of the operating regimes of the closed loop con-trollers. This method uses as main tool the phase trajectory of the control error, which is analysed in a qualitative manner. The operating domain is divided into four regimes: variable, steady, oscillating and unstable. This operation may be performed by fuzzy or by interpolative controllers, on the base of general knowledge on the PID controllers adjustment. A high quality self-adaptation of the controllers may thus be obtained, covering all the possible operation regimes.

Keywords: *fuzzy interpolative controllers, adaptive control, phase trajectory, heuristic control rules.*

-

1. INTRODUCTION

The control of the non-linear and time variable plants demands a high quality self-adaptation, covering a wide range of possible combinations of the parameters of the system. A conventional control fails when it has to deal with contradictory clauses. For instance the integrative effect of the PID controller may produce notable overshoots and oscillations during the vari-able regimes but in the same time its help is welcome during the steady ones. The adjustment of a dc drive controller for ordinary speeds does not matches at low speeds, because of to the non-linearity of the friction load torque, which is growing when the speed is decreasing.
However these contradictions may be managed by the help of the fusion of several controllers each one designed for a specific operating regime. The fuzzy fusion of the controllers became in the last years a reliable solution to this problem.
The goal of this lesson is to offer a simple and reliable method for the on-line identification of each regime that could possible produce contradictions when adjusting a PID basic controller. The lesson is based on a previous paper [8].

2. THE BASIC ASSUMPTIONS

The first question that comes in our mind when dealing with barely controllable plants (highly nonlinear, death time, mathematically unknown, etc.) is: “how would a human operator control this plant?” Many specialists may disregard this heuristic approach, but still it has fundamental advantages over the numerical algorithms: it can always be applied! Of course the performances may be not optimal, and a rigorous solution to the stability problem is not possible, but in turn the costs of the development of new products is getting lower and special applicative measures against instability can always be considered. And nevertheless, the accumulation of experimental facts about the heuristic controlled systems is a basic condition for the developing of further numerical algorithms.

The basic assumptions that we will take in account are the next ones:

- The PID is the fundamental control algorithm since, in a general way of speaking, it can handle the present (P), the history (I) and the future (D) of the evolution of the system. The expert knowledge of the PID adjustment offers reliable control rules for the most part of the possible control applications.
- The self-adaptation is strictly necessary for highly nonlinear and/or time variable plants.
- A very attractive adaptation tool for the on-line control is the 2D phase trajectory of the control error $\dot{\boldsymbol{\varepsilon}} = f(\boldsymbol{\varepsilon})$ (the dependence between the change of error $\dot{\boldsymbol{\varepsilon}}$ and the error $\boldsymbol{\varepsilon}$). The phase trajectory of the error is a fundamental tool, which has a significant weight in the elaboration of the control decisions by the human operators.
- The adaptive strategy will be a heuristic one, the only one that is able to cope with the highly nonlinear systems.
- The fuzzy logic is a basic tool that allows us to cope with the specific incertitude of the complicated nonlinear systems and with the qualitative or heuristic approaches [6], [7].
- The fuzzy fusion of the controllers or only of the adaptive part of the controllers is necessary for the management of the contradictions.
- The fuzzy controllers may be developed by a linguistic method. In order to obtain reliable implementations we will use only fuzzy controllers that have a linear interpolation correspondent (so called fuzzy-interpolative controllers). The prod-sum Sugeno fuzzy inference and COG defuzzyfication is an obvious first choice [3].

3. THE *FFSAIC* ADAPTIVE CONTROLLERS

Any fuzzy controller is an interpolative one as well, and may be implemented by means of a look-up table with linear interpolation (similar to the Matlab-Simulink look-up table). Such an implementation may be considered as a *fuzzy interpolative controller* [3]. The main advantage of this kind of structures consists in the easiness of the implementations (both software or hardware). Interpolative controllers are able to perform quite similarly to any other kind of controllers having in the mean time the advantage of a low amount of calculations and of the speed [4], [5]. The electronic implementations are feasible in any possible technology, even in the analogical ones, since the only important mathematic operation involved is the linear interpolation.

Yet the look-up tables are strictly numerical tools, their representation in the human mind being inadequate, especially when using large or multidimensional tables. Thus the fuzzy feature becomes useful mainly for methodological reasons. The linguistic representation of the knowledge is revelatory for humans, catalysing the developing stages of the applications. The fuzzy controllers used in this paper will be fuzzy-interpolative, and by consequence they will be implemented by look-up tables.

We will consider as a recommendable tool a specific controller structure which is operating by analysing the phase trajectory of the error and which was designed having in mind the previous assumptions. This structure will be called FSAIC (*Fuzzy Self-Adapted Interpolative Controller*) [3] and it is shown in Fig.1.

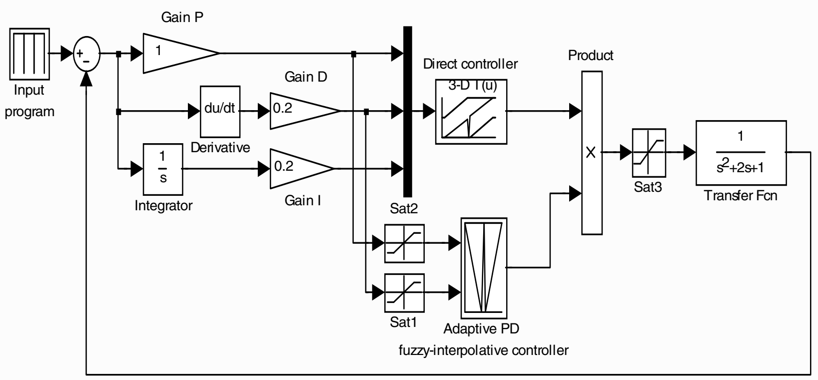


Fig. 1. A FSAIC controller

The characteristic features of this structure are the following:

● *FSAIC has a variable structure*. During transient regimes the main controller is a PD one. During the steady regime an integrative effect is gradually introduced, the structure becoming a PID one. This functionality may be achieved with a 3D look-up table having as inputs $\boldsymbol{\varepsilon}$, $\dot{\boldsymbol{\varepsilon}}$ and $\int\boldsymbol{\varepsilon}$, the integrative of the error. The different PD tables corresponding to the $\int\boldsymbol{\varepsilon}$ dimension differ only at the central rule, that is activated when $\boldsymbol{\varepsilon}$ = **zero** and $\dot{\boldsymbol{\varepsilon}}$ = **zero** [1], [2], [3]. Thus the integrative effect is gradually activated, through a linear interpolation, only when steady regimes occur. The block that fulfils this functionality in Fig. 1 is Direct controller.

● *A fuzzy-interpolative PD controller (corrector) induces the adaptive feature*. The adaptive controller is generating a multiplying correction (*Gain*) over the output of the main controller; the multiplying correction is preferable to the additive one by allowing a direct fuzzy fusion. The PD structure is chosen because it can be matched with the phase trajectory of the error (see Table 1, next page). The corresponding block in Fig. 1 is Adaptive PD fuzzy-interpolative controller.

In the following paragraphs we will focus on the adaptive correction. The adaptive control rules could be grouped into four clusters according to the next classification of the main operating regimes: variable (G1), steady (G2), oscillating (G3) and unstable (G4). This point of view is not the only possible, other classifications being as well productive. The clusters of rules must respect in great shapes the next linguistic commitments:

- G1: < *Gain* is **medium** and *Integrative* is **zero** >
- G2: < *Gain* is **great** and *Integrative* is **great** >
  (1)
- G3 & G4: < *Gain* is **small** and *Integrative* is **zero** >

The *Integrative* variable is generated by Direct controller.

The differences between G3 and G4 are not fundamental. If necessary they may be separated by the help of supplementary criteria, for example with the help of the product of the first and second derivatives of the control error, that is positive for unstable systems [3]. Anyway, *Gain* must be reduced in order to reject the oscillations and/or to stabilize the system, in the sense of the Nyquist stability criterion. Due to the non-linearity of the plant, an optimal crisp value of the *Gain* would have no sense.

In the next table a possible structure of adaptive PD fuzzy-interpolative corrector is presented.

Table 1: An adaptive PD fuzzy-interpolative controller

change of error $\dot{\varepsilon}$ / **error ε**	**negative**	**zero**	**positive**
positive big	**G1**	**G1**	**G1**
positive small	**G3**	**G3**	**G3**
zero	**G1**	**G2**	**G1**
negative small	**G3**	**G3**	**G3**
negative big	**G1**	**G1**	**G1**

A typical phase trajectory of the error is underlying its the correspondence with the table.

4. THE FUZZY FUSION OF THE ADAPTIVE CORRECTORS

When the blending of the rules is not possible or satisfactory, the fundamental solution is the fuzzy fusion of the individual controllers. The simplest fuzzy fusion operates according to the weighted sum formula:

$$u(t) = \frac{\sum_i \mu_i(t) \cdot u_i(t)}{\sum_i \mu_i(t)} \tag{2}$$

where u_i is the output of the controller i and μ_i is the membership value of the same controller. More complicated shapes for the membership values functions may be used when imposed performances must be reached.

A *Fusioned Fuzzy Self-Adapted Interpolative Controller* (FFSAIC) is presented in Fig. 2.

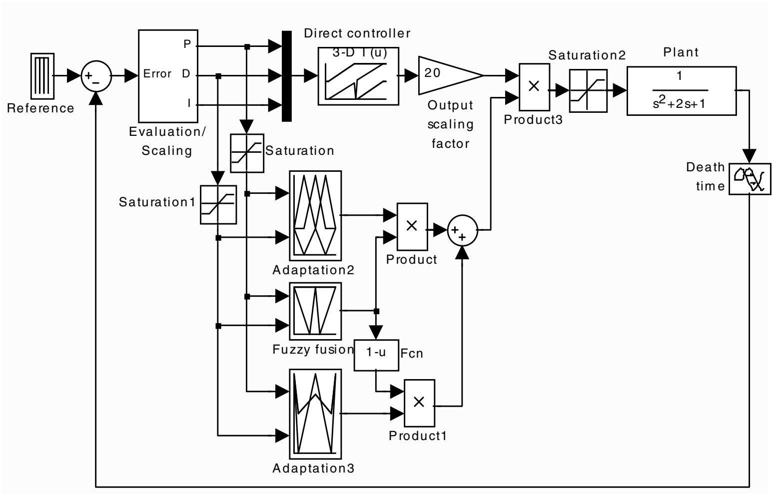


Fig. 2. A FFSAIC controller

FFSAIC may include several different PD adaptive controllers (correctors), each one dedica-ted to a specific operating regime. In Fig. 2 a minimum FFSAIC variant is presented, having only two correctors, corresponding to the regimes G1&G2 (which may be covered by the same adaptive corrector) and G3. The Table 2 controller may be applied in order to control the fuzzy fusion of three correctors. The heuristic meanings of the identifications of G1 and G3 are obvious, while the identification of G3 (oscillations) is linked to the points $\dot{\boldsymbol{\varepsilon}} = \mathbf{0}$: if the error is **zero** (in the linguistic sense) the regime is steady, if not, the regime is oscillatory.

Table 2: A regime identification fuzzy-interpolative controller

change of error $\dot{\varepsilon}$ / error ε	negative	zero	positive
positive big	G1	G1	G1
positive small	G1	G3	G1
zero	G1	G2	G1
negative small	G1	G3	G1
negative big	G1	G1	G1

5 COMPARING *FSAIC* TO LINEAR *PID*. SIMULATIONS RESULTS

The next examples have only an illustrative goal. Some detailed results may be found in [3].

Fig. 3 presents a comparison of FSAIC and linear PID performances for the case of an oscillatory plant, with the transfer function

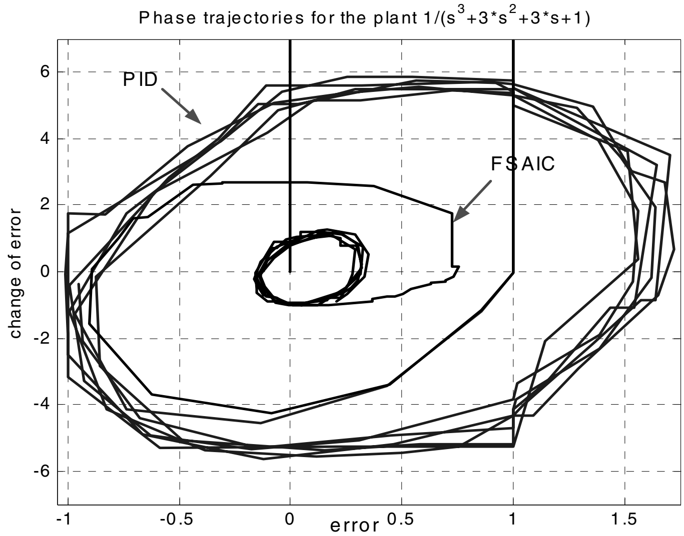


Fig. 3.The performance of FSAIC comparing to PID

$$H(s) = \frac{1}{s^3 + 3 \cdot s^2 + 3 \cdot s + 1} \tag{3}$$

Fig. 4 presents the time performances of FFSAIC when controlling four different plants:

$$H_1(s) = \frac{1}{s^2 + 2 \cdot s + 1} \tag{4}$$

$$H_2(s) = \frac{e^{-0.025s}}{s^2 + 0.2 \cdot s + 1} \tag{5}$$

$$H_3(s) = \frac{e^{-0.025s}}{s^2 + 20 \cdot s + 1} \tag{6}$$

$$H_4(s) = \frac{e^{-0.025s}}{s^3 + 3 \cdot s^2 + 3 \cdot s + 1} \tag{7}$$

If linear PID controllers would control them, the tested plants would produce extremely different responses, one of them being fully unstable.

6 CONCLUSIONS

The identification of the operating regime of the closed loop control systems may be obtained by the means of the *qualitative* analyse of the phase trajectory of the control error. Adaptive actions based on this approach are able to improve the control of highly nonlinear and/or important dead times plants.

This analysis may be achieved with the help of fuzzy-interpolative controllers. A family of fuzzy self-adaptive interpolative controllers FSAIC is designed to implement this kind of operation. The adaptive part of FSAIC is a fuzzy-interpolative PD corrector.

The regime identification may ensure a high quality adaptation even in the case of contradictory clauses. The tool that can cope with the contradictory possible regimes (FFSAIC) is obtained by the fuzzy fusion of several adapting FSAIC correcting controllers. The on-line control of the fuzzy fusion is achieved as well by a fuzzy-interpolative controller.

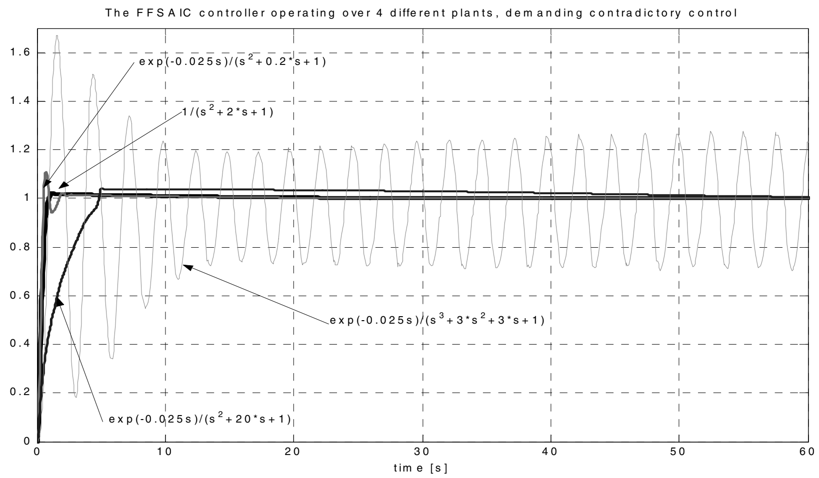


Fig. 4. Tne FFSAIC with the on-line identification of the operating regime

FFSAIC can also control systems that are suffering unstabilising influences.

Only two possible applications of FSAIC are revealed so far: the air-conditioning [1] and the ABS braking [2] of the railway coaches. Further research could produce important applicative achievements, since the method is a versatile and easy to implement one.

REFERENCES:

[1] Bălaş M., The temperature control by variable structure interpolative PID controllers. *Proceedings of the 6th International Conference on Engineering of Modern Electric Systems*. University of Oradea, May, 2001.

[2] Bălaş M., Bălaş V., Foulloy L., Galichet S., A model of the sliding wheel during braking. *Proceedings of the 5th International Conference on Railway Bogies and Running Gears BOGIE'01*, Budapest, Sept., 2001.

[3] Bălaş M., Regulatoare fuzzy interpolative, Politehnica Timişoara, 2002.

[4] Dragomir T.L., Interpolative Controllers – a Correctional and Improvement Alternative for Design and Implementation. *Proceedings of the 6th International Conference on Engineering of Modern Electric Systems EMES'01*, Oradea, May, 2001.

[5] Dragomir T.L., Dale S., Bălaş M., Some aspects regarding interpolative control. *The 13th International Conference on Control Systems, CSCS13*, Bucureşti, Nov. 2001.

[6] Foulloy L., Qualitative Control and Fuzzy Control: Towards a Writing Methodology, *AICOM* Vol. 6, Nrs. 3/4 Sept./Dec., pag. 147-154, 1993.

[7] Stoll K.E., Ralston P.A.S., Ward T.L., Linguistic Design of Nonlinear Controllers. *Journal of Intelligent and Fuzzy Systems*, vol. 4, nr.1, 19-32, 1996.

[8] M. Bălaş, V. Bălaş, T.L. Dragomir, „A Preamble on the Identification of the Operating Regimes of the Controllers by the Help of the Phase Trajectory”, Electronic BUSEFAL nr. 87, June 2002, http://www.univ-savoie.fr/ Portail/ Groupes/ LISTIC/ busefal/ Papers/ 87.zip/ 87_01

Constantin VOLOŞENCU

Universitatea „Politehnica” Timişoara
B-dul Vasile Pârvan Nr. 2,
300223 Timişoara, Romania
Email:
constantin.volosencu@aut.upt.ro

FAULT DETECTION AND DIAGNOSIS IN INDUSTRIAL SYSTEMS, BASED ON FUZZY LOGIC – A SHORT REVIEW

Note: *The paper was presented to the International Summer School „SOFT COMPUTING” Arad-Moneasa 16.08 – 22.08.2004*

Abstract

The paper presents a fuzzy method in fault detection and diagnosis. This method provides a systematic framework to process vague variables and vague knowledge. The supervision of the process requires the treatment of quantitative and qualitative knowledge. Here fuzzy logic approaches are especially attractive for symptom generation with fuzzy thresholds, linguistically described observed symptoms and the approximate reasoning with multi levele fuzzy rule based systems for fault symptom tree structures.

Keywords: *process monitoring*, *fuzzy logic, fault detection, diagnosis*

1. INTRODUCTION

1.1. General Things

In the field of control engineering new topics continue to flourish and develop. In common with general scientific investigations the new concept of fault detection and process monitoring emerge as a new growth area in process control. This innovative concept coalesces into a new sub-discipline. A little maturity has been acquired by the new concept. Archival publications and monographs may be study in the international scientifically literature. The reason of developing fault detection and diagnosis is based on new real time instrumentation technologies in manufacturing plant installations. Process operators are using process data to minimize plant downtime and optimize plant operations. The traditional routes to fault detection were model based and them the process has to be well understood. An alternative group of methods has emerged which do not require the use of an explicit model. Model-free and non-parametric methods for fault detection, process optimization and control design are in current development.

Many technical processes are operating now under automatic control based on the progress of control theory. A general scheme for the organization of process automation tasks in several levels is presented in fig. 1 [Ise95].

The measurement and manipulation takes place at in the lowest level. The next level contains feedforward and feedback control. Here some variables u, y are adjusted according to conditions v or reference variables w. The second level contains supervision; it is the monitoring level, which serves to indicate undesired or unpermitted process (fault detection) states and to take appropriate actions (based on fault identification and diagnosis) as process recovery, fail-safe, shut-down, triggering of redundance systems or reconfiguration schemes. The higher levels may perform tasks such as optimization, coordination or general management in order to meet economic demands or

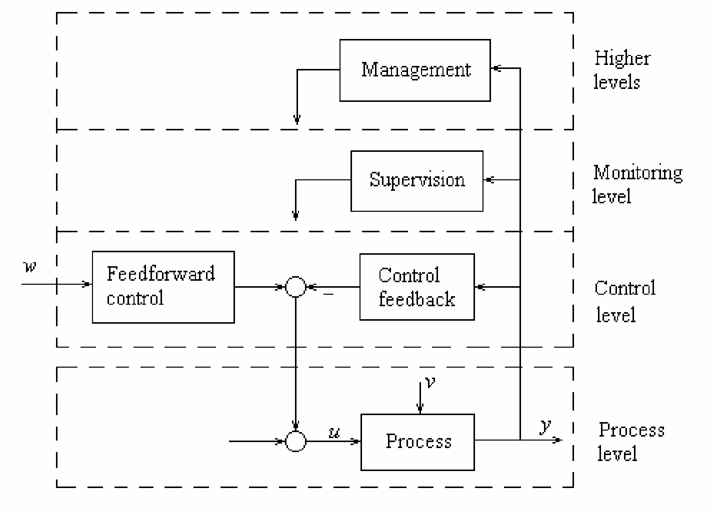


Fig. 1. The block diagram of multilevel process automation

scheduling. The lower levels need a fast reaction and act locally and the higher levels are dedicated to tasks that act globally.

Modern manufacturing facilities are large scale, highly complex and operate with a large number of variables under closed-loop control. An early accurate fault detection and diagnosis for these plants can minimize downtime, increase safety of plant operations and reduce manufacturing costs. Plants are becoming more heavily instrumented, resulting in more data becoming available for use in detecting and diagnosing faults. Classical methods have a limited ability to detect and diagnose faults in such multivariable processes. This has led to a surge of scientifically effort to develop more effective process monitoring methods. To apply these methods in the real industrial systems a large amount of research must be done.

The domain of control of electrical drives is a large file of application for the fault detection and diagnosis techniques. This domain is appropriate for practicing engineering in process monitoring. Numerous and ample simulators for a large scale electrical drives applications may be developed. Data from the process – electrical drive – may be collect and applied to the monitoring techniques to detect, isolate and diagnose various fault. The process monitoring techniques can be modeled, simulated and implemented using commercial software as Matlab and Simulink [Bea94, Che92, Lot93, Vol2'0/1, 2].

These researches are focused on the approach based on fuzzy logic and neural networks, as new concepts developed versus the conventional concepts as statistical quality control, analytical methods, knowledge base methods, canonical variant analysis or Fisher discriminate analysis.

In the control of the electrical drives there is a large push to assure higher security of the electrical drive, to reduce electrical machine rejection rates and to satisfy the increasing stringent safety of the entire process controlled by the electrical drives and its environment. To meet the higher standards, the electrical drives contain a large number of variables operating under closed-loop control. The standard process controllers, for example PID controllers, are designed to maintain satisfactory operations by compensating for the effects of disturbances and changes occurring in the process. While these controllers can compensate for many types of disturbances, there are changes in occurring in the electrical drives that the controllers cannot handle adequately. These changes are called *faults*. More precisely, a fault is defined as a no permitted deviation of at least one characteristic property or variable of the system (resistance, inductance, moment of inertia, friction coefficient, voltage, flux or current, or quality criteria, as for example the overshoot).

The types of faults occurring in industrial systems include process parameter changes, disturbance parameter changes, actuator problems and sensor problems. Short circuit of the motor resistance or friction coefficient rising are examples of process parameter changes. A disturbance parameter change can be an extreme change of in the load torque or in the ambient temperature. An example of an actuator problem is a destroyed transistor in the power converter, and a sensor producing biased measurements is an example of a sensor problem. To ensure that the process operations satisfy the performance specifications, the faults in the process need to be detected, diagnosed and removed. These tasks are associated with *process monitoring*. Different methods were developed for process monitoring.

The goal of process monitoring is to ensure the success of the planned operations by recognizing anomalies of the behavior. The information not only keeps the plant operator and maintenance personnel better informed of the status of the process, but also assists them to make appropriate remedial actions to remove the abnormal behavior from the process. As a result proper process monitoring, downtime is minimized, safety of plant operations is improved and manufacturing costs are reduced. As industrial systems have become more highly integrated and complex, the faults occurring in modern processes present monitoring challenges that cannot be treated with conventional methods. The weakness of the linear methods of detection and diagnosis have led to a surge of research concentrated to the methods of the artificial intelligence as fuzzy logic and neural networks. The growth of the research activity can also be explained by the fact that industrial systems are becoming more heavily instrumented, resulting in larger quantities of data available for use in process monitoring and that modern computers are becoming more powerful. The availability of data collected during various operating fault conditions is essential to process monitoring. The storage capacity and computational speed of modern computers enable process monitoring algorithms to be computed when applied to large quantities of data [Chi01].

The electrical drives, controlled in real time with digital equipments and software of high performances, are at this moment suitable for the implementation of such monitoring algorithms [Har94].

1.2. Process Monitoring Procedures

The four procedures associated with process monitoring are [Chi01]: *fault detection, fault identification, fault diagnosis and process recovery.*

Fault detection is determining whether a fault occurred. Early detection may provide invaluable warning on emerging problems, with appropriate actions taken to avoid serious process upsets.

Fault identification is identifying the observation variables most relevant to diagnosing the fault. The purpose of this procedure is to focuses the plant operator's and engineer's attention on the subsystems most pertinent to the diagnosis of the fault, so that the effect of the fault can be eliminated in a more efficient manner.

Fault diagnosis is determining which fault occurred, in other words, determining the cause of the observed out-of-control status. It can be more specifically defined as determining the type, location, magnitude and time of the fault. The fault diagnosis procedure is essential to the counteraction or elimination of the fault.

Process recovery, also called *intervention*, is removing the effect of the fault and it is the procedure needed to close the *process monitoring loop*. Whenever a fault is detected, the fault identification, fault diagnosis and process recovery procedures are employed in the respective sequence, otherwise, only the fault detection procedure is repeated.

The schema of the process monitoring is presented in fig. 2. Whenever a fault is detected, the fault identification, fault diagnosis and process recovery procedures are employed in the respective sequence, otherwise only the fault detection procedure is repeated.

It is not necessary to implement all four procedures in a process monitoring. For example, a fault may be diagnosed (the procedures of fault diagnosis) without identifying the variables immediately affected by the fault (the procedure of fault identification) and recover the normal operation. Often, the goal of process monitoring is to incorporate the plant operators and engineers into the process monitoring loop efficiently rather then to develop an entire automat monitoring scheme.

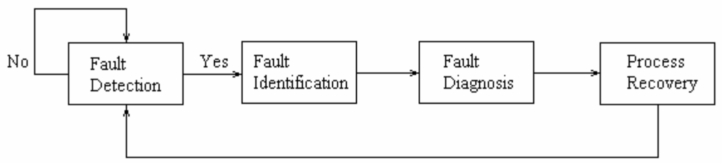


Fig. 2. The schema of the process monitoring loop

After a fault occurs, the process may be recovered, reconfigured or repaired and returning to the control strategy. Once a fault has been properly diagnosed the optimal approach to counteract the fault may not be obvious. A feasible approach may be the return to the standard process control strategy. Several methods were developed to evaluate the controller performance and these can be used to determine to which control strategy need to be returned to restore the satisfactory performances. For example, in the case of a sensor problem, a sensor reconstruction technique can be applied to the process to restore the control operations. Even the process recovery is an important and necessary part of the process monitoring loop, it is not the focus of this approach. All the phases presented in the above scheme may be implemented on the control systems of the electrical drives.

1.3. Process Monitoring Measures

The process surveillance contains one or more measures, based on different theories, as: statistics, pattern classification and system theory. These measures represent the behavior of the process. The idea is to convert on-line data collected from the process into a few measures and to assist the operators in determining the status of the operations and diagnosing the faults. For fault detection limits must be placed on the measures. A fault is detected whenever one of the evaluated measures is outside the imposed limits. In this way, a measure is able to define a wrong behavior of the process accordingly the out of control status. In these measures the values of the variables can be compared with the values of other variables to determine the variable most affected by the fault. Developing and comparing measures that accurately represent the different faults of the process can also diagnose faults.

The goal of process monitoring is to develop measures that are maximally *sensitive* and *robust* to all possible faults. Faults are manifested in several ways and no the all faults may be detected and diagnosed with only a few measures. Each measure characterizes a fault in different manner; one measure will be more sensitive to certain faults and less sensitive to other faults relative to other measures. This motivates using multiple process monitoring measures, with the proficiency of each measure determined for the particular process and the possible faults.

Possible monitoring measures can be classified as being associated with one or more of approaches [Chi01]: *data driven, analytical* and *knowledge-based*. The data driven measures are derived directly from process data. Modern industrial systems as the electrical drives are large-scale systems. They are equipped with a lot of instruments, which produce large amount of data. Too much information is beyond the capability of an operator to assess the process operations from observing the data. The data driven techniques have the ability to transform the high-dimensional data into a lower dimension, in which the important information is captured. The improvement of a large-scale process monitoring scheme may be done using statistical techniques. The disadvantage of the data driven techniques is that their proficiency is highly dependent on the quantity and quality of the process data.

The analytical approach uses mathematical models often constructed from the primary mathematical and physical laws of the processes. The analytical approach is applicable to information rich systems, with available satisfactory models and enough sensors. Most analytical measures are based on parameter estimation, observer based design and parity relations. The applications of the analytical approach are made on systems with a small number of inputs and outputs and states. Because the analytical approaches need detailed models in order to be effective it is difficult to apply these measures to large-scale systems. Whenever, the analytical measure may be apply to the electrical drives, because they have a small number of inputs and outputs and also they have good state space models. The main advantage of the analytical approach is the ability to incorporate physical understanding of the process into the monitoring strategy. In other words, when detailed models are available, the analytical measures have good performances. The electrical drives are systems suitable for application of the analytical measures. Models of the electrical are developed.

The knowledge-based approach uses qualitative models to develop process monitoring measures. The knowledge-based approach is well suited for systems with non-available detailed models, with complex and nonlinear models. The knowledge-based measures are based on causal analysis, expert systems or pattern recognition. Like the analytical approach the applications of the knowledge-based measures are on system with a small number of inputs, outputs and states. The electrical drives are non-linear systems with complex models. For an easier application of the knowledge-based measures software packages must be developed.

1.4. Process Monitoring Methods

1.4.1. Fault Detection Methods

Different m*odel based fault detection methods* were developed in the last 25 years, by using mathematical model. The tasks consist in the detection of faults in the technical process, actuators and sensors by measuring the available input and output variables *u* and *y*. Fig. 3 presents the scheme of such a detection in a control system.

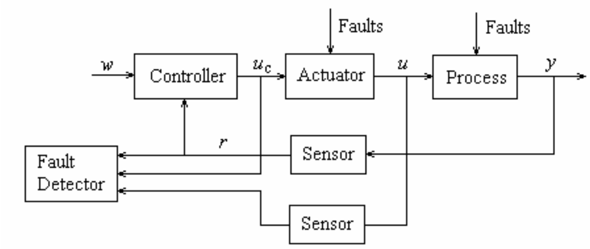


Fig. 3. The scheme of fault detection in a control system

The process is considered to operate inn open loop. A distinction can be made between static and dynamic, linear and nonlinear process models.

Several important model based fault detection methods are used. *Parameter estimation* presents changes of parameter estimates. *State estimation* presents the changes of states estimates or output errors. *Parity equations* offer the output error or polynomial error.

The measured or estimated quantities like signals, parameters, state variables, or residuals are usually stochastic variables with mean value and variance, as normal values for the non-faulty process. Analytical symptoms are obtained as changes with reference to the normal values. If a fixed threshold is used, a compromise has always to be made between the detection of small faults and false alarms because of short term exceeding. Methods of change detection, which estimated the change of the mean, value and relate it to the standard deviation may improve the decision. Membership fuzzy logic functions may be used to specify the universe of discourse for a such approach.

1.4.2. Fault Diagnosis Methods

The task of fault diagnosis consists of determining the type of fault with as much possible such as the fault size, location and time of detection. The diagnosis procedure is based on the observed symptoms of the process.

Based on heuristic knowledge and inference mechanism for diagnosis e*xpert systems* are used to imitate the reasoning of human experts when diagnosing faults. The experience from a domain can be formulated in terms of rules, which can be combined with the knowledge from the physical-mathematical laws that are ruling the process or with a structural description of the system. Expert systems are able to capture human diagnostic reasoning that are not expressed into mathematical models.

Pattern recognition techniques use associations between data patterns and fault classes without an explicit modeling of internal process states or structures. Examples are *artificial neural networks* and *self organizing maps*. These techniques are related to the data driven techniques in terms of modeling the relation between data pattern and fault classes. Neural networks are black boxes that learn the pattern entirely from the training sessions [Chi01].

All methods based on data-driven, analytical and knowledge based approaches have their advantages and disadvantages, so that no single approach is best for all applications. Usually the best process monitoring scheme uses multiple statistics or methods for fault detection, identification and diagnosis. A good approach is to incorporate several techniques for process monitoring as neural network with fuzzy logic and expert systems. This can be beneficial also in the application of the electrical drives.

2. FUZZY LOGIC APPLICATIONS FOR PROCESS MONITORING

2.1. General Things

Many control systems are based today on fuzzy logic. The development of the fuzzy control theory offers alternative techniques for process monitoring strategies, which describes human reasoning in linguistic form. The fuzzy supervision knowledge-based may replace a human operator. Fuzzy logic provides a systematic framework to process vague variables and knowledge. There are some automation functions for which the fuzzy logic may offer attractive possibilities and advantages.

The main elements in the process information, as: input variables, the process classes, the automation functions and the output variables have different degrees of vagueness. The crisp information may be treated with fuzzy logic.

With regard to the *input signals* usual measurement equipment is designed to deliver signals with precise mean values and low standard deviations. *Low cost sensors* may have imprecise outputs with biased values and large standard deviations. *Non directly measurable variables* can only be calculated or estimated based on other measurement variables and analytical models and therefore imprecise to some extent. Also *linguistic values* of a human operator may be processes. Hence, the degree of precision and imprecision of the input variables may vary within fairly broad ranges.

The description of the static and dynamic behavior in form of a mathematical models plays a crucial role as well for the design of the technical processes as for a systematic design of high performance control systems. Processes from different process classes (mechanical, electrical, thermal, chemical and biological) may be described using basic physical- mathematical principles. Theoretical modeling can be applied using energy balance equations, state equations and phenomenological laws. The structure of the mathematical model is obtained with lumped parameter or distributed parameter (ordinary or partial differential equations). By this way quantitative models based on physical laws result and one can distinguish the following cases [Ise95]. *Analytical process models*, they are quantitative models. If the structure is known, the parameters can be determined experimentally by parameter estimation and identification methods. Quantitative process models result from proper combination of theoretical and experimental modeling. If physical laws in the form of equations cannot express the internal behavior, as for not well-defined process, some qualitative information on the causalities may be expressed as *rules*: *If <condition> then <conclusion>*. The condition part contains as inputs the memberships of facts to the premise and possibility their logical connection by *AND*, *OR* and *NOT*. The conclusion part describes the logical consequence on the output. Then one can distinguish h*euristic process models* (qualitative models) [Ise95].

The functions of the process automation (*control, supervision, management, man-machine interface*) are usually organized according to the levels shown in fig. 1. For *supervision* the basic task is to check if variables exceed tolerances. These tolerances are usually a compromise between small values to early detect faulty behavior and large values to avoid false alarms subject to normal process fluctuations. In reality these tolerances are not crisp but fuzzy variables. Moreover, in many cases fault diagnosis means to treat vague process knowledge and therefore to apply rules [Chi01, Ise95].

The man-machine interface is designed for human interaction. Basic tasks of the human operator are higher-level tasks as supervision, management and redundancy for all control functions. The actions of human operators are based on qualitative knowledge and many be better describes by rules than by precise algorithms. The information in process automation generally changes from using precise algorithms at the lower levels to using more rules at the higher levels.

Actuators with electrical, pneumatic or hydraulic drives need crisp inputs. *Output variables* to the human operators through the man-machine interface can be either crisp.

2.2. Knowledge based fault detection

Within automatic control of technical systems supervisory functions serve to indicate undesired or unpermitted process states and to take appropriate actions in order to maintain the operation and to avoid damages or accidents. Following functions can be distinguished [Ise95]:
Monitoring: measurable variables are checked with regard to tolerances and alarms are generated for the operator;
Automatic protection: in the case of dangerous process state, the monitoring function initiates automatically an appropriate counteraction;
Supervision with fault diagnosis: based on measured variables features are calculated, symptoms are generated via change

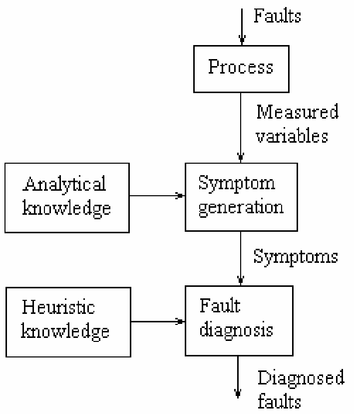


Fig. 3. Scheme of knowledge-based fault detection and diagnosis

detection, a fault diagnosis is performed and decisions are made for counteractions.

The diagram of knowledge based fault detection and diagnosis is presented in fig. 4. The main tasks can be subdivided in fault detection by symptom generation and fault diagnosis. The symptom generation can be *analytic* or *heuristic*.

Analytical symptom generation is based on analytical knowledge on the process, used to produce quantitative analytical information. Based on measured process variables a data processing has to be performed to generate first characteristic values by: -*Limit value* checking of directly measurable signals. Characteristic values are exceeding signal tolerances. –*Signal analysis* of directly measurable signals by use of signal models like correlation functions, frequency spectra, autoregressive moving average models. Characteristic values are amplitudes, frequencies or model parameters. –*Process analysis* by using mathematical process together with parameter estimation, state estimation and parity equation methods. Characteristic values are parameters, state variables or residuals. These features are compared with the normal features of the non-faulty process and *methods of change detection and classification* are applied. The resulting changes of the described direct measured signals, signal models or process models are then *analytical symptoms* of faults.

Heuristic symptom generation is in addition to the symptom generation with quantifiable information and they can be produce by using *qualitative information* from human operators. Through human observation and inspection heuristic characteristic values in form of special noise, colour, smell, vibration, wear and tear etc. are obtained. The process history in form of performed maintenance, repair, former faults and lifetime, load measures constitute a further source of heuristic information. Statistical data achieved from experience with the same or similar processes can be added. By this way *heuristic symptoms* can be represented as linguistic variables (small, medium, large) or as vague numbers (around a certain value).

An example for a stochastic symptom treated with fuzzy logic is presented in fig. 5.

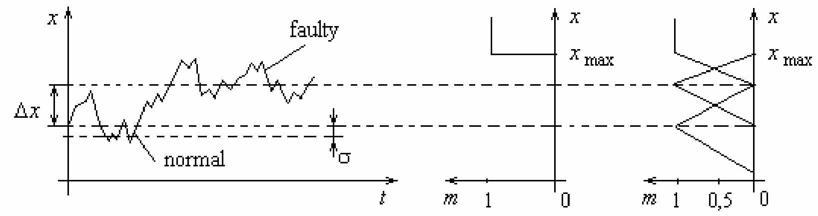


Figure 5. Fuzzy membership approach for a stochastic variable *x*(*t*)

A fixed threshold x_{max} is used and a compromise between the detection of small faults and false alarms must be made. A triangle membership function $m(x)$ is specified with the center as the mean and the lower width due to a Gaussian distribution 68% or 95% of all values lie within those intervals. A combined symptom representation of the mean and the standard deviation is obtained. By matching the current value $m(x)$ with the symptom's membership function we obtain a gradual measure for exceeding a *fuzzy threshold*.

For the processing of all symptoms in the inference engine it is advantageous to use a unified representation. One possibility is to present the analytic and heuristic symptoms with confidence numbers $c(x)\in[0, 1]$ and treatment in the sense of probabilistic approaches known from reliability theory […]. Another possibility is the representation as membership function $m(x)\in[0, 1]$ of fuzzy sets (fig. 6).

By these fuzzy sets and corresponding membership functions all analytic and heuristic symptoms can be represented in a unified manner within the range [0, 1]. These integrated symptoms are the inputs for the inference mechanism, presented in the following chapter.

For establish heuristic knowledge bases for diagnosis there are several approaches. In general specific rules are applied in order to set up logical interactions between observed symptoms (effects) and unknown faults (causes). The propagation from appearing faults to observable symptoms in general follows physical cause-effect relationships where physical properties and variables are connected to each other quantitatively and also as functions of time. The underlying physical laws are frequently not known in analytical form or too complicated for calculations. Therefore heuristic knowledge in form of qualitative process models can be expressed in form of IF-THEN rules: *IF* (*premise*) THEN (*conclusion*).

The condition part (premise) contains facts in the form of observed symptoms Δx as input and the conclusion part includes events and faults as a logical cause of the facts. This procedure results in fault symptom trees, relating symptoms to events and faults. Then symptoms or events are associated by AND and OR operators.

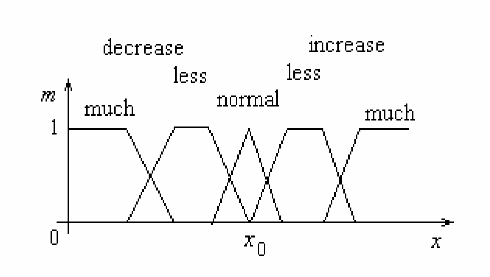


Fig. 6. Examples of membership functions for symptoms

Based on the available heuristic knowledge in form of heuristic process models and weighting of effects different diagnosis forward and backward reasoning strategies can be applied. Finally the diagnosis goal is achieved by a fault decision, which specifies the type, size and location of the fault as well as its time detection. By using the strategy of forward chaining a rule, the facts are matched with the premise and the conclusion is drawn based on the logical consequence. Therefore with the symptoms Δx as inputs the possible faults are determined using the heuristic causalities. In general the symptoms have to be considered as uncertain facts. Therefore a representation of all observed symptoms as membership function $m(x)$ of fuzzy sets in the interval [0, 1] is feasible, especially in unified form.

3. THE BASIC STRUCTURE OF A FUZZY ENGINES FOR FAULT DIAGNOSIS

A *fuzzy inference engine* has the basic structure from fig. 7.
It contains the *fuzzification* and *defuzzification* interfaces and the inference based on *fuzzy rules*. The input *u* and output *y* information is crisp. The fuzzification interface offers a fuzzy information *A* of the inputs *u*. The inference offers also a fuzzy information *B* for the outputs *y*. The inputs *u* are symptoms and the outputs *y* are diagnosis of faults.

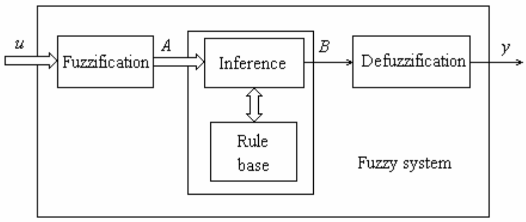


Fig. 7. The structure of a fuzzy system for fault diagnosis

An example of an *inference* based on *min-max method* is presented in fig. 8.

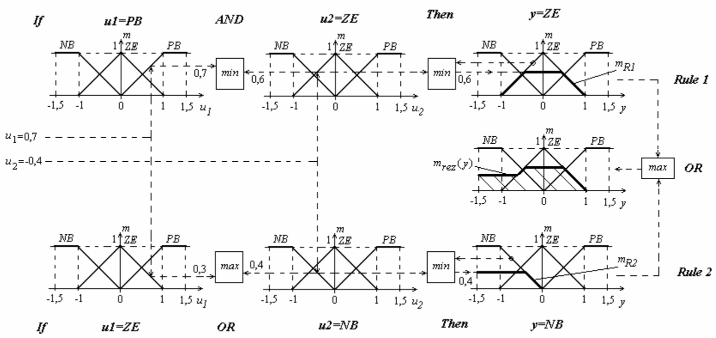


Fig. 8. The min-max inference

The most used defuzzification is based on the *center of gravity method*, presented in fig. 9.

Approximate reasoning with fuzzy logic is made with the structure from fig. 7 of a fault symptom tree. A fuzzy rule based

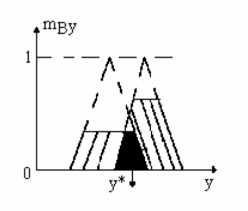


Fig. 9. Defuzzification

system with multiple levels of rules can be established. The symptoms u=Δx are represented by fuzzy sets A, with linguistic meanings like "*normal N*", *"less increased LI*", "*much increase MI*" etc.

In contrast with fuzzy control *fuzzy fault diagnosis* differs by:

- inputs are mostly no crisp measurements, but detected symptoms represented as fuzzy sets;
- not only one level or rules does exist, but mostly several levels;
- frequently it is difficult to specify the membership functions of the intermediate events because of very vague knowledge.

The approximate reasoning follows the steps describe in fig. 7: fuzzyification, rule activation with evaluation by the compositional rule of inference and defuzzification.

The dimension of the fuzzy rule base is given by: number of symptoms, number of rules per level, number of levels and number of faults.

The overall dimension may therefore blow up strongly even for small componets or processes. Therefore the software implementation is important. Mainly two procedures to perform the reasoning are known: *sequential rule of activation* and *multiple rule activation* [Ise95].

Top reduce the computation effort simplifying assumptions may help.

4. CONCLUSION

In the fault detection and diagnosis fuzzy logic provide a systematic framework to process vague variables and vague knowledge. As the vagueness in fault detection is high the potential of fuzzy logic grow to higher levels.

The supervision of the process requires the treatment of quantitative and qualitative knowledge. Here fuzzy logic approaches are especially attractive for symptom generation with fuzzy thresholds, linguistically described observed symptoms and the approximate reasoning with multi levele fuzzy rule based systems fro fault symptom tree structures.

REFERENCES

[Bea94] Beale, M. - *Neural Network Toolbox for Matlab*, MathWorks, Inc., 1994.

[Cha97] Chang, C.S.; Chen, J.M.; Liew, A.C.; Srinivasan, D.; Wen, F.S. – Fuzzy Expert System for Fault Diagnosis in Power Systems, *Int. Journal of Engineering Intelligent Systems for Electrical Engineering & Communications,* 1997.

[Che92] Chekoway, C.; Kirk, K.; Sullivan, D. - *Simulink User's Guide*, Mathworks, Inc., 1992.

[Chi01] Chiang, L.H.; Russell, E.L; Braatz, R.D. – *Fault Detection and Diagnosis in Industrial Systems*, Springer, London, 2001.

[Har94] Harashima, F. - *Power Electronics and Motion Control-A Future Perspective*, Proceedings of the IEEE, Nr. 8, 1994.

[Hua98] Huallpa, B.N.; Nobrega, E.; Zuben, F.J.V. – Fault Detection in Dynamic Systems Based on Fuzzy Diagnosis, *Proc. of the IEEE Int. Conf. On Fuzzy Systems*, Piscataway, N.J., 1998.

[Ise95] Iserman, R. – On Fuzzy Logic Applications for Automatic Control, Supervision and Fault Diagnosis, *The Proceedings of EUFIT'95*, Aachen, 1995.

[Lot93] Lotfi, A. - *Fuzzy Inference Systems Toolbox for Matlab*, MathWorks, Inc., 1993.

[Vol99] Voloşencu, C. – Consideraţii privind utilizarea reţelelor neuronale în identificarea sistemelor neliniare, *Revista Română de Informatică şi Automatică,* Vol. 9, nr. 3, 1999.

[Vol2'0/1] Voloşencu, C. – Bibliotecă de programe pentru sisteme fuzzy, *Conf. de informatică teoretică şi tehnologii informaţionale CITTI'2000*, Constanţa, 2000.

[Vol2'0/2] Voloşencu, C. - Bibliotecă de programe pentru reţele neuronale*, Analele Universităţii "Aurel Vlaicu" din Arad*, 2000.

[Zim91] Zimmermann, H.J. – *Fuzzy Set Theory and Its Applications*, Kluwer Academic Publishers, Boston, 1991.