

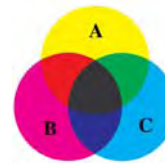
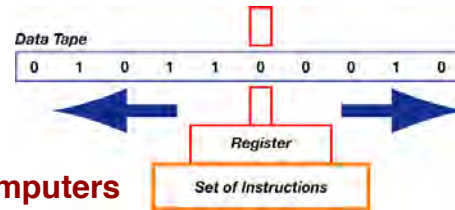
Computers, Computing, and Sets

Robert Stengel

Robotics and Intelligent Systems MAE 345,
Princeton University, 2015

Learning Objectives

- Principles of Turing Machine
- Evolution of Calculators and Computers
- Programming
- Crisp Sets
 - Properties of collections of objects
- Fuzzy Sets and Control Systems
 - Effects of uncertainty or imprecision



Copyright 2015 by Robert Stengel. All rights reserved. For educational use only.
<http://www.princeton.edu/~stengel/MAE345.html>

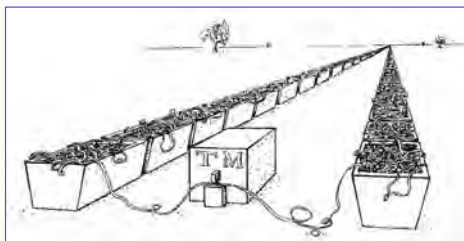
1

Turing Machine (Logical Computing Machine)



Alan Turing
IAS, Princeton, *38

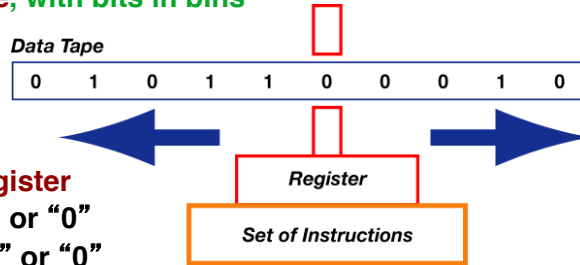
- Abstract representation of programming for a computing device
 - Attempt to give mathematically precise definition to algorithm or mechanical (or effective) procedure
 - Hardware description as a machine is figurative
- Finite number of internal discrete states of the machine
 - States = steps or instructions of a program
- Unlimited amount of external input data on a tape



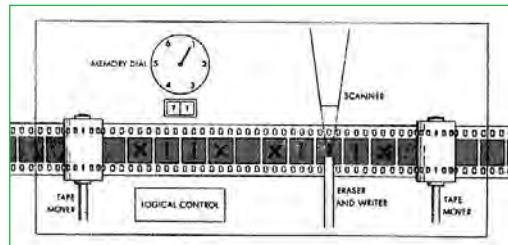
2

Elements of a Logical Computing Machine (LCM)

- **Data Tape**, with bits in bins



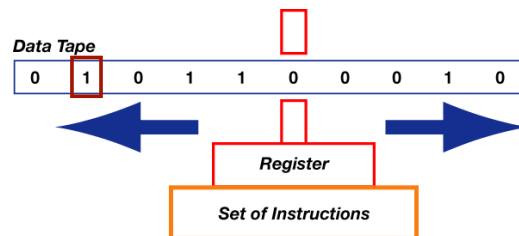
- **Read/Write Register**
 - Senses “1” or “0”
 - Assigns “1” or “0”
- **Set of Instructions (Program) for R/W Register**
 - Define **internal states**, which are identified by **state (i.e., program line) number**
- **Control Mechanism**
 - **Move read/write head** over tape
 - **Or move tape** through read/write head



3

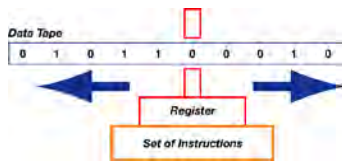
Operation of a Turing Machine

Execution begins in field of “0”s to the left, with device at **State #0**



- **State #0 instruction** moves device **to the right** until it encounters a “1”
- **Action of device** depends on current instruction in a Stored Program
 - **Modifies the bin** (or not)
 - **Identifies next internal state**
 - **Moves one bin** to right or left

4



Example of a Turing Machine: Denary (Base 10) Notation for State (Line Number)

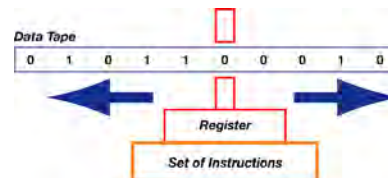
- Execution begins in field of "0"s to the left, with device at **State #0**
- **State #0** instruction moves device **to the right** until it encounters a "1"
- Action of device depends on stored instructions
 - Modifies the bin (or not)
 - Identifies next internal state
 - Moves one bin to right or left

Penrose, 1989

[Program] Instruction State #	[Data] Register Contents	Next State	New Bin Contents	Direction of Move
0	0	0	0	R
0	1	13	1	L
1	0	65	1	R
1	1	1	0	R
2	0	0	1	R (Stop)
2	1	66	1	L
3	0	37	1	L
...
210	0	3	1	L
...
258	1	0	0	R (Stop)
259	0	97	1	R
259	1	0	0	R (Stop)

5

Same Example: Binary-coded Program Line Number and Move Direction



[Program] Instruction State #	[Data] Register Contents	Next State	New Bin Contents	Direction of Move
0	0	0	0	1
0	1	1101	1	0
1	0	1000001	1	1
1	1	1	0	1
10	0	0	1	1 (Stop)
10	1	1000010	1	0
11	0	100101	1	0
...
11010010	0	11	1	0

- Execution begins in field of "0"s to the left, with device at State #0
- **State #0** instruction moves device to the right until it encounters a "1"
- Action of device depends on stored instructions
 - Modifies the bin (or not)
 - Identifies next internal state
 - Moves one bin to right or left

6

Unary, Binary, and Expanded Binary Coding of Data

- Turing Machine to add "1" to a unary number (at right)

- On data tape: 01111

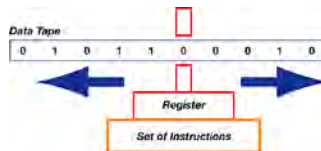
[Program] Instruction State #	[Data] Register Contents	Next State	New Bin Contents	Direction of Move
0	0	0	0	R
0	1	1	1	R
1	0	0	1	R (Stop)
1	1	1	1	R

- Binary coding problems

- Notation for terminating binary description of a number
- Definition of space between numbers
- Recognition of **de-limiters** (e.g., commas) and **logical/arithmetic operators**

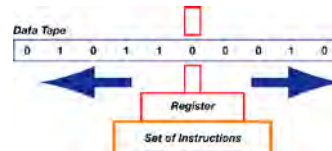
- Data could be stored in **unary format**

- On data tape: 01110



7

Unary and Expanded Binary Coding of Data



- **Solution: Expanded Binary Coding**

- **Contraction**: Unary expression of small numbers, separated by "0"
- **Encode** de-limiters and operators as numbers
- **Terminate** numbers with commas

Number	Meaning	Expanded Binary Notation
0	Binary "0"	0
1	Binary "1"	10
2	Comma	110
3	Minus Sign	1110
4	Plus Sign	11110

8

Expanded Binary Coding

Expanded Binary Code: 0 1 0 0 0 1 0 | 1 1 0 1 0 1 0 | 1 0 0 0 1 1 1 0 | 1 0 1 0

 | | | | | | | | | | | | | | | |

Number (code) 1 0 0 1 2 1 1 2 1 0 0 3 1 1

Meaning (de-code) 9 , 3 , 4 - 3

Number	Meaning	Expanded Binary Notation
0	Binary "0"	0
1	Binary "1"	10
2	Comma	110
3	Minus Sign	1110
4	Plus Sign	11110

9

Expanded Binary Coding and Turing Machines

- Expression of arbitrary denary statement numbers
 - Convert from denary to binary
 - Convert from binary to expanded binary
- Define Turing Machine for operations on expanded binary code

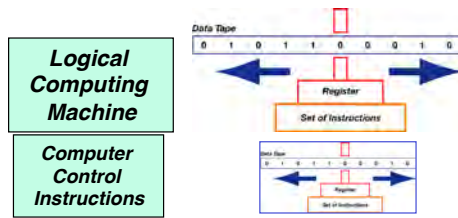
Original: 0 1 13 1 L

Binary: 0 1 1101 1 0

De-Limited: ,0,1,1101,1,0,

Expanded Binary 000110110101101010010110110110

10



Further Evolution of the Turing Machine

- **Universal Turing Machine (Computer Program in “Machine Language”)**
 - Turing Machine for **control** of a Turing Machine
 - **Instructions on a separate tape** or at beginning of data tape
 - **2nd TM** with **simple code set** reads register contents
 - **Instruction tape could be modified** just like the data tape
 - Instruction branches can be conditioned on prior results
 - **Self-modifying instruction** set possible
- **Church-Turing Thesis (“Hypothesis”, “Conjecture”)**
 - Turing Machine (**LCM**) **defines what we mean** by an algorithmic, mechanical, effective, or recursive procedure
 - **LCM** can do anything that could be described as a **Rule of Thumb** or “**purely mechanical**”

11

Calculation and Computing

12

Calculation



- **Thinking,** augmented by
 - Abacus
 - Slide rule
 - Math tables
 - Mechanical calculator

Computer



13

Early Computers - IBM

026 Key Punch



**Punch Card
One line of code)**



IBM 7094



http://en.wikipedia.org/wiki/IBM_7094

IBM 360/91



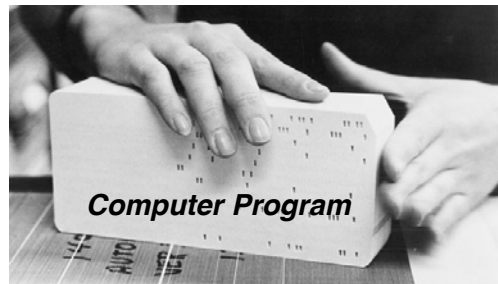
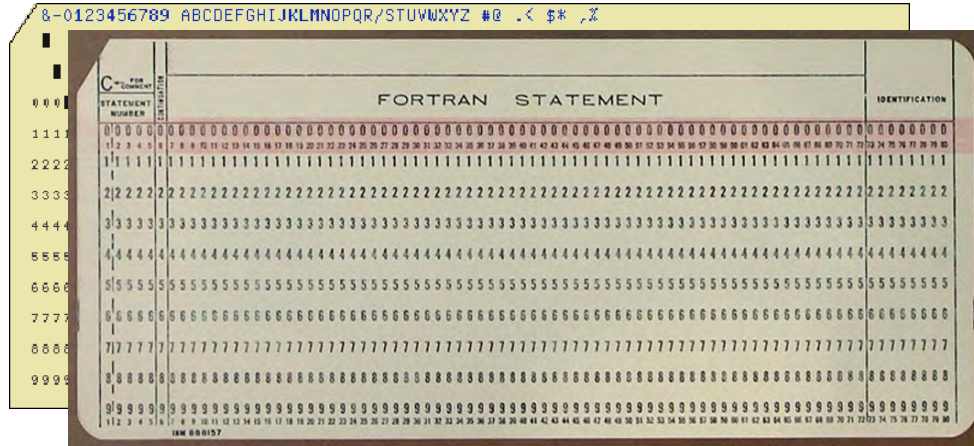
IBM 650 Computer

http://en.wikipedia.org/wiki/IBM_650

http://en.wikipedia.org/wiki/IBM_360

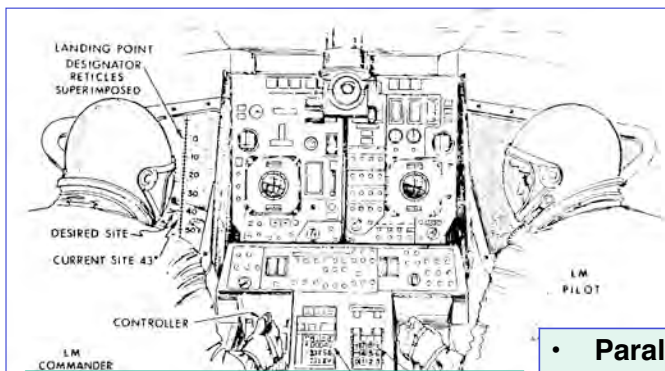
14

Early Computers – Punched Card



15

Apollo Guidance Computer

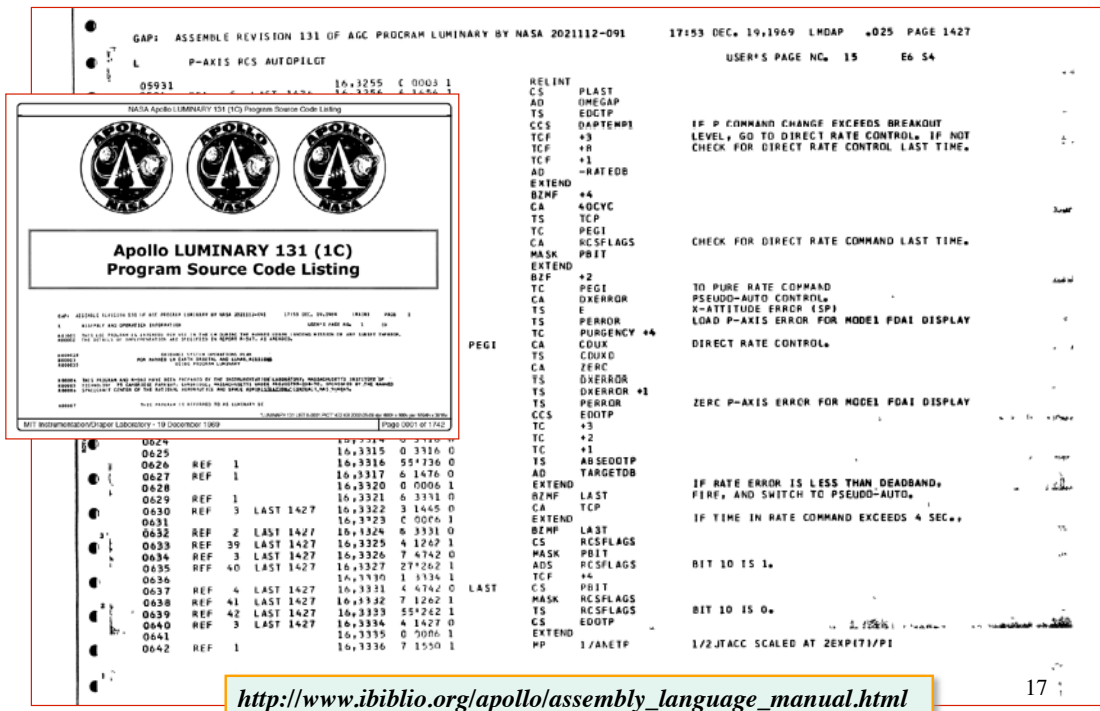


- Parallel processor
- 16-bit word length (hexadecimal)
- Memory
 - 36,864 words (fixed)
 - 2,048 words (variable)
- 1st operational solid-state computer
- Identical computers in CSM and LM
 - Different software (with many identical subroutines)

http://klabs.org/history/build_agc/

https://en.wikipedia.org/wiki/Apollo_Guidance_Computer

A Little AGC Digital Autopilot Code



Evolution of Programming

- History of programming languages
 - <http://www.levenez.com/lang/history.html>
- The song, “99 Bottles of Beer on the Wall”, programmed in 1,500 computer languages
 - <http://www.99-bottles-of-beer.net/>
 - **In BASIC:**

```
10 REM BASIC Version of 99 Bottles of beer
20 FOR X=100 TO 1 STEP -1
30 PRINT X;"Bottle(s) of beer on the wall,";X;"bottle(s) of beer"
40 PRINT "Take one down and pass it around,"
50 PRINT X-1;"bottle(s) of beer on the wall"
60 NEXT
```

Programming Language Classes

- Expert systems can be programmed in almost any language
- Language is the interface between the programmer and the computer
- **Higher-order**
 - **Lower-order**
 - Interpreter
 - Compiler
 - Assembly language
 - » Machine language
- **Critical differences**
 - Instruction set
 - Execution speed
 - Memory use
- **Ultimately, it is all machine code**

- **Procedural**
 - Imperative
 - Functional
- **Non-Procedural Languages**
 - Declarative
 - Non-Declarative

19

Crisp Sets

20

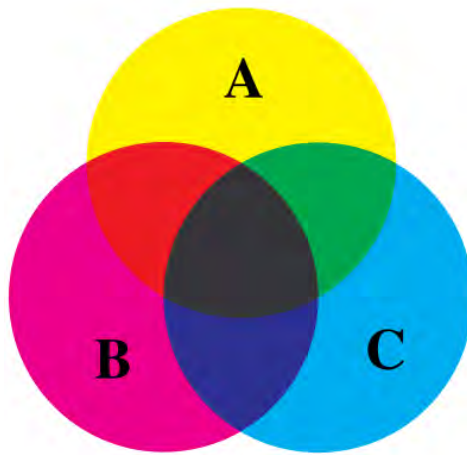
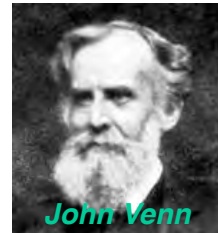
Naive (or Intuitive) Set Theory (1870s)



- Deals with the properties of well-defined collections of objects
- **Universal set = Universe of discourse = U**
 - Contains all elements of possible concern in a particular context
- **A = a particular set in U**
 - defined in a **list**
 - by a **rule**, or
 - by a **membership function** describing elements (or members) of the set

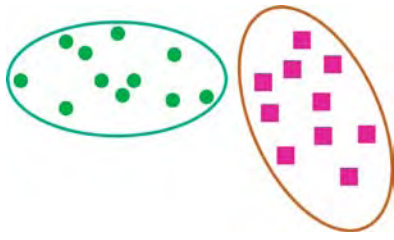
21

Venn Diagrams (1881)



- **A:** All mammals
- **B:** All aquatic animals
- **C:** All gray, hairless objects
- **A & B:** Whales, dolphin, seals, ...
- **B & C:** Fish, clams, whales, dolphins, ...
- **A, B, & C:** Whales, dolphins, ...

22



Set Theory

- **List** $A = \{x, y, z, \dots\}$ Ordering of elements is **not** important
 $A = (x, y, z, \dots)$ Ordering of elements **is** important
- **Rule** $A = \{x \in U \mid x \text{ meets some conditions}\}$ *e.g.*,
 $A = \{x \in U \mid g(x) \leq 0\}$
- **Membership function** $\mu_A(x)$

$$\mu_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases}$$

http://en.wikipedia.org/wiki/Table_of_mathematical_symbols

23

Membership in a Set

- **A** = a particular set in **U**
 - defined in a list or rule, or
 - by a membership function describing elements (or members) of the set
- Universal set = guests at a party
- Particular sets
 - Current graduate students
 - Alumni
 - Spouses
 - Friends of students
 - Children
 - Same family
 - Visitors
 - Pilots
 - Teachers
 - Managers
 - Military officers
 - Women and men
 - US citizens or foreign nationals

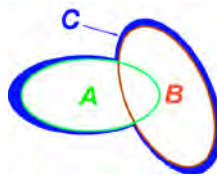


24

Operations on Sets

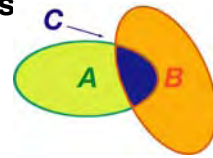
- Union of sets

$$C = A \cup B$$



- Intersection of sets

$$C = A \cap B$$



- One-to-one correspondence

$$A = B$$

For example, as when

$A = (x, y, z)$

$B = (4, 3, 9)$; then $x = 4$, $y = 3$, and $z = 9$

- Proper Subset

$$B \subset A$$

$$B \subseteq A, \text{ and}$$

$$B \neq A$$



... in which case, A "is greater than" B

$$A \supset B$$

25

Properties of Sets

- Complement

$$A' = U - A$$



- Empty (null) set

$$\emptyset = A - A \text{ or } = U - U$$

- Reflexive property

- Relationships that bear same effect on own set as on other sets

$$\text{e.g., } A = A, A \geq A, A \leq A$$

- Symmetry property

- Relationship of first to second set is the same as second to first set

$$\text{e.g., } A = B, B = A$$

- Transitive property

- Two sets bear same relationship to a third set

$$\text{e.g., if } A > B \text{ and } B > C, \text{ then } A > C$$

- Equivalence

- Reflexivity + Symmetry + Transitivity

$$A \sim B \text{ or } A \equiv B$$



26



What If Sets Have Uncertain or Contradictory Membership?

- Example: U = All the cars in Berkeley

$$A = \{x \in U \mid x \text{ has 4 cylinders}\}$$

$$B = \{x \in U \mid x \text{ has 6 cylinders}\}$$

Set

A

B

C

D

Crisp Logic

4-cylinder engine

6-cylinder engine

8-cylinder engine

Others

Set

E

F

Fuzzy Logic

US cars

Foreign cars

- What makes a car “US”?
 - US cars may contain foreign parts
 - Foreign cars may contain US parts
- How should we define Sets E and F ?

List?
Rule?
Membership function?

27

Fuzzy Sets

28

Hard and Soft Thinking

- Problem-solving approaches

- Logical / Metaphorical
- Reasonable / Dream-like
- Serious / Humorous
- Definite / Ambiguous
- Consistent / Paradoxical
- Laborious / Playful
- Exact / Approximate
- Real / Fantastic
- Focused / Diffuse
- Analytical / Illogical
- Specific / General
- Mature / Immature

- Crisp / Fuzzy



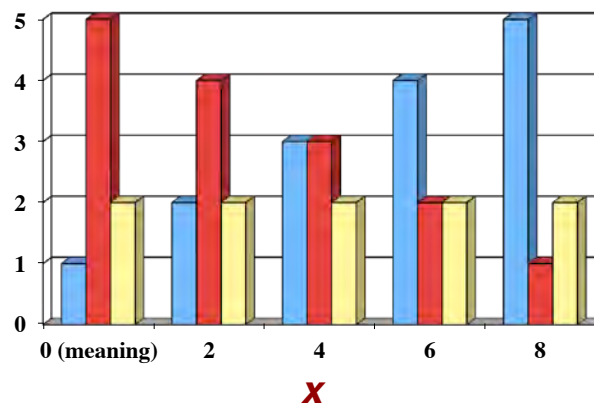
29

A Notional Fuzzy Experiment

- What does each term mean?

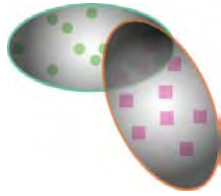
A few $\leq x$
A lot $\geq x$
Several $\approx x$

of People
who Agree



Normalize results so that the maximum is 1
Normalized plots are fuzzy membership functions

30



Fuzzy Sets

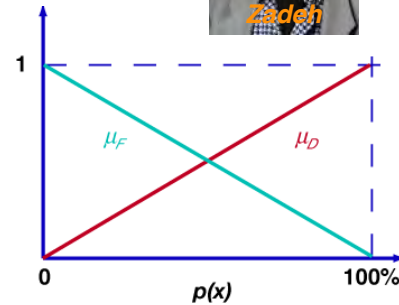


- **Fuzzy membership function**, $\mu_A(x) = 1$
 - takes any value in $[0, 1]$

- **Fuzzy set**, E

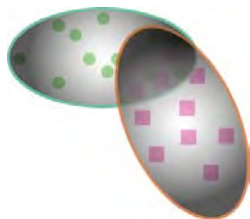
$$E = \{x \in U \mid [x, \mu_E(x)]\}$$

- U = All the cars in Berkeley
 - $p(x)$ = percentage of domestic parts = $\mu_D(x) = 1$
 - $1 - p(x)$ = percentage of foreign parts = $\mu_F(x) = 1$



- Membership functions express a **subjective utility** that may be rigorous (e.g., based on **probability**) or not
- “If 46% of a car’s parts are domestic, it probably is a foreign car”

31

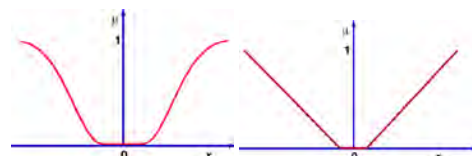


Fuzzy Membership Functions

“Close to zero”



“Probably not y”



“About y”



“Young/old”



32

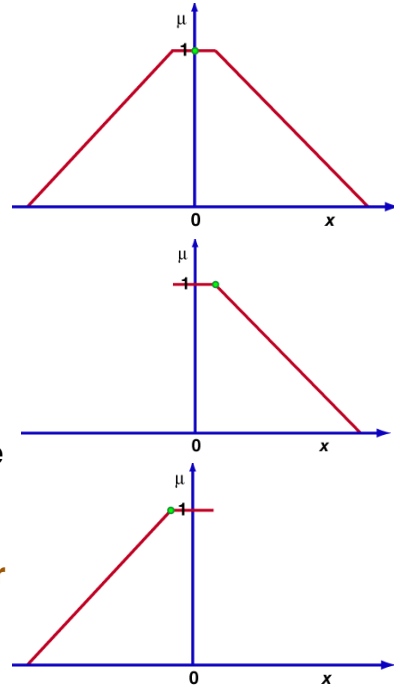
Fuzzy Set Definitions

- **Support of a fuzzy set**

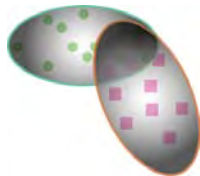
$$\text{supp}(A) = \{x \in U \mid \mu_A(x) > 0\}$$

- **Center of a fuzzy set**

- **Two-sided:** center = mean value
- **Left-sided:** center = max (x) for which $\mu(x) = 1$
- **Right-sided:** center = min (x) for which $\mu(x) = 1$



33



Fuzzy Logic Operations

Union of sets

$$A \cup B$$

if and only if $\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \forall x \in U$

Intersection of sets

$$A \cap B$$

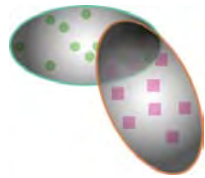
if and only if $\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \forall x \in U$

A contains B

$$A \supset B$$

if and only if $\mu_A(x) \geq \mu_B(x) \forall x \in U$

34



More Fuzzy Logic Operations

Equivalence of sets

$$A \sim B$$

if and only if $\mu_A(x) = \mu_B(x) \forall x \in U$

Complement of sets

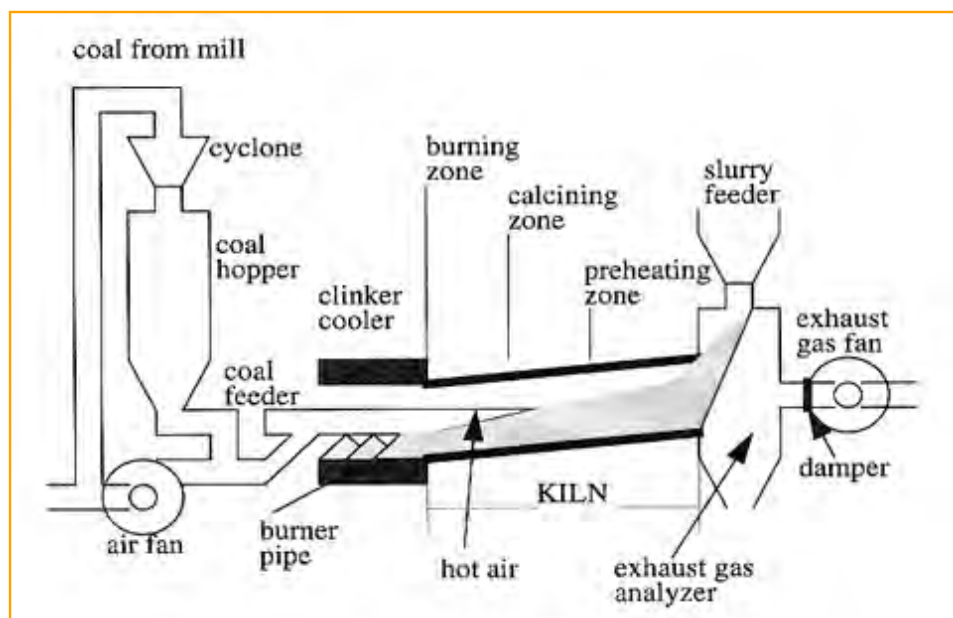
$$A' = U - A$$

if and only if $\mu_{A'}(x) = 1 - \mu_A(x) \forall x \in U$

Fuzzy logic is a *generalization* of crisp logic based on the definition of the *membership function*

35

Example: Cement Kiln Control

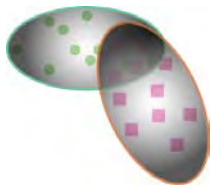


36

Cement Kiln Operator's Manual

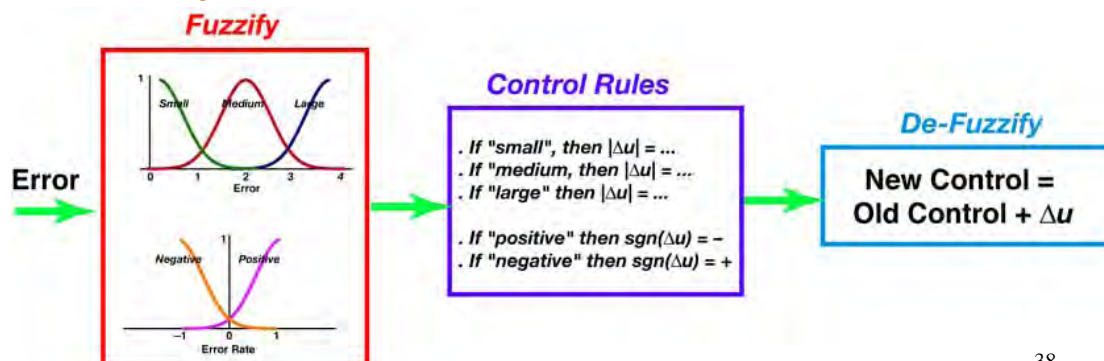
Case	Condition	Action to be taken	Reason
10	BZ ok	a. Increase air fan speed	To raise back-end temperature and increase oxygen for action 'b'
11	OX low BE low BZ ok	b. Increase fuel rate a. Decrease fuel rate speed slightly	To maintain burning zone temperature To raise oxygen
12	OX low BE ok BZ ok	a. Reduce fuel rate b. Reduce air fan speed	To increase oxygen for action 'b' To lower back-end temperature and maintain burning zone temperature
13	BE high BZ ok OX ok	a. Increase air fan speed b. Increase fuel rate	To raise back-end temperature To maintain burning zone temperature
14	BE ok BZ ok OX ok	None. However, do not get overconfident and keep all conditions under observation	
15	BE ok BZ ok OX ok BE high	When oxygen is in upper part of range a. Reduce air fan speed When oxygen is in lower part of range b. Reduce fuel rate c. Reduce air fan speed	To reduce back-end temperature To raise oxygen for action 'c' To lower back-end temperature and maintain burning zone temperature
16	BZ ok OX high	a. Increase air fan speed b. Increase fuel rate	To raise back-end temperature To maintain burning zone temperature and reduce oxygen
17	BE low BZ ok OX high BE ok	a. Reduce air fan speed slightly	To lower oxygen

37



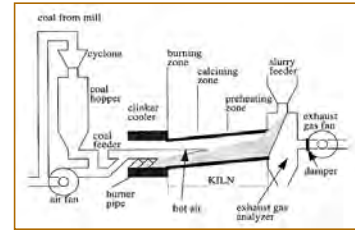
Linguistic (Mamdani) Fuzzy Control Systems (after Schramm)

- **Antecedent** and **consequent** are both fuzzy propositions
 - e.g., “If **error** is **small** and **error rate** is **negative**, then **control command** is **small**”
 - What are “small”, “medium”, and “large”?
- Must “**fuzzify**” physical error/rate, apply **fuzzy rules**, and “**de-fuzzify**” control command



38

Mamdani Fuzzy Controller for Cement Kiln (from Wang)



Linguistic Controller

Antecedents

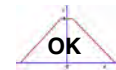
- **BZ**: Temperature in burning zone
- **OX**: Oxygen in exhaust gas
- **BE**: Temperature at end of kiln

Consequents

- **CR**: Coal feed rate
- **DP**: Exhaust damper position

27 fuzzy rules, e.g.,

- If BZ is OK and OX is low and BE is low, then set CR to large, and DP to large

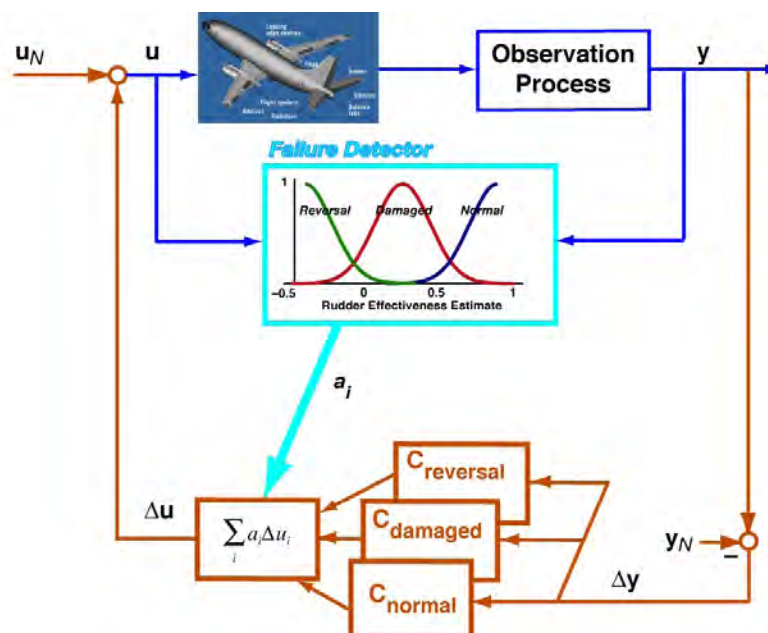


Controller is apparently symbolic, but symbols must have values for computation, i.e., Fuzzy Membership Functions

39

Gain-Scheduling (Takagi-Sugeno) Fuzzy Control Systems

(Schramm, Gopisetty, and Stengel, 1998)

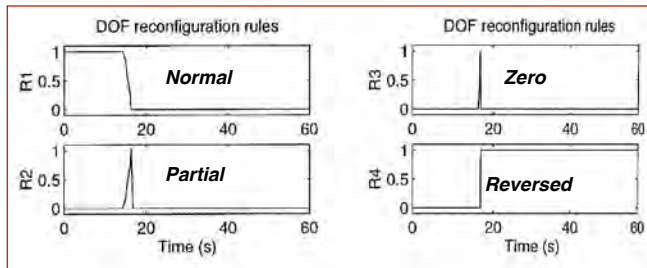


40



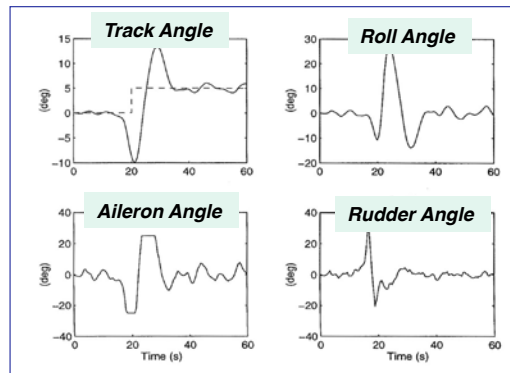
Rudder Reversal Simulation with Failure-Tolerant Fuzzy Control System

- Rudder reversal occurs at $t = 10$ s
- Heading angle change commanded at $t = 20$ s



- Failure detection logic detects nothing until rudder effect is expected
- Once detected, control signal is reversed

Schramm, 1998



41

Fuzzy Logic \neq Fuzzy Thinking

- Quantitative approach to reasoning under uncertainty
- “**Possibility theory**” vs. **Probability theory** (Lotfi Zadeh, 1978)
- Relationship to other **uncertainty belief systems** of artificial intelligence, e.g.,
 - Bayesian belief network
 - Dempster-Shafer theory
 - Transferable belief model
 - Certainty factors
- Propositions are true or false only within the **context of a paradigm**

*1998 CDC debate: Zadeh vs. Athans
re: fuzzy vs. conventional control*



*Michael
Athans*

42

***Next Time:
Probability and Statistics***

43

***Supplemental
Material***

44

Turing Machine for Euclid's Algorithm

Penrose, R., *The Emperor's New Mind*, 1989, p. 41

Since we wish to be able to include numerical data as part of our input, we shall want to have a way of describing ordinary numbers (by which I here mean the natural numbers 0, 1, 2, 3, 4, . . .) as part of the input. One way to do this might be simply to use a string of n 1s to represent the number n (although this could give us a difficulty with the natural number zero):

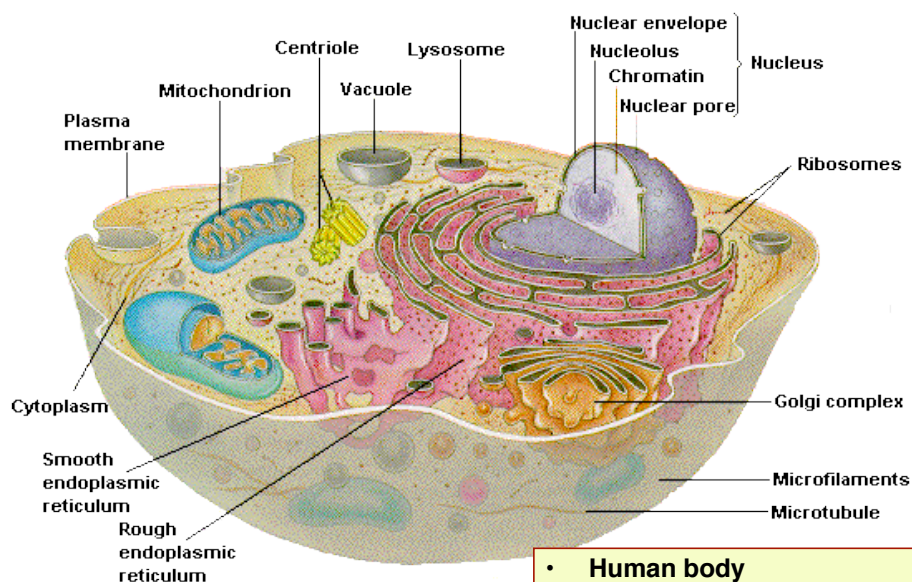
$1 \rightarrow 1, 2 \rightarrow 11, 3 \rightarrow 111, 4 \rightarrow 1111, 5 \rightarrow 11111, \text{ etc.}$

This primitive numbering system is referred to (rather illogically) as the *unary* system. Then the symbol 'O' could be used as a space to separate different numbers from one another. It is important that we have such a means of separating numbers from one another since many algorithms act on *sets* of numbers rather than on just single numbers. For example, for Euclid's algorithm, our device would need to act on the *pair* of numbers A and B. Turing machines can be written down, without great difficulty, which effect this algorithm. As an exercise, some dedicated readers might perhaps care to verify that the following explicit description of a Turing machine (which I shall call EUC) does indeed effect Euclid's algorithm when applied to a pair of unary numbers separated by a O:

$00 \rightarrow 00R, 01 \rightarrow 11L, 10 \rightarrow 101R, 11 \rightarrow 11L, 100 \rightarrow 10100R,$
 $101 \rightarrow 110R, 110 \rightarrow 1000R, 111 \rightarrow 111R, 1000 \rightarrow 1000R, 1001 \rightarrow 1010R,$
 $1010 \rightarrow 1110L, 1011 \rightarrow 1101L, 1100 \rightarrow 1100L, 1101 \rightarrow 11L, 1110 \rightarrow 1110L,$
 $1111 \rightarrow 10001L, 10000 \rightarrow 10010L, 10001 \rightarrow 10001L, 10010 \rightarrow 100R,$
 $10011 \rightarrow 11L, 10100 \rightarrow 00STOP, 10101 \rightarrow 10101R.$

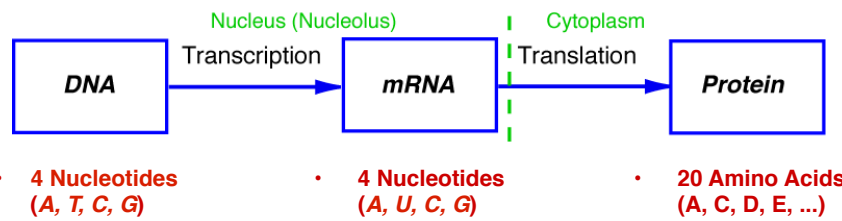
45

Turing Machines in Biological Cells



46

The Central Dogma: Core Process of Protein Production



- A **gene** is a sequence of **nucleotides** on a **chromosomal strand of DNA**
- **Single-stranded messenger RNA (mRNA)** expresses information from DNA to form a **protein**, a sequential molecule of **amino acids**
- Information coded in **nt** triplets (**codons**), with **open reading frames** defined by **Start** and **Stop** codons

U A C G G C

The Genetic Code					
First Position	Second Position		Third Position		
U	U	C	A	G	U
	F	S	Y	C	C
	L	S	Stop	Stop	A
	L	S	Stop	W	G
C	L	P	H	R	U
	L	P	H	R	C
	L	P	Q	R	A
	L	P	Q	R	G
A	I	T	N	S	U
	I	T	N	S	C
	I	T	K	R	A
	I	T	K	R	G
G	V	A	D	G	U
	V	A	D	G	C
	V	A	E	G	A
	V	A	E	G	G

47

System View of The Core Process



- **Polymerase unzips** dual-stranded DNA
- **Exons of DNA** code mRNA sequence with help of **transcription factors**
- **Introns** and **extragenic DNA** are ignored (**deleted**) in RNA
- mRNA enters **ribosome**, which is composed of **ribosomal RNA (rRNA)** and **proteins**
- **Transfer RNA (tRNA)** transports amino acids to ribosome
- **Amino acids** assembled into **protein** according to **RNA sequence** (3 nt per codon)

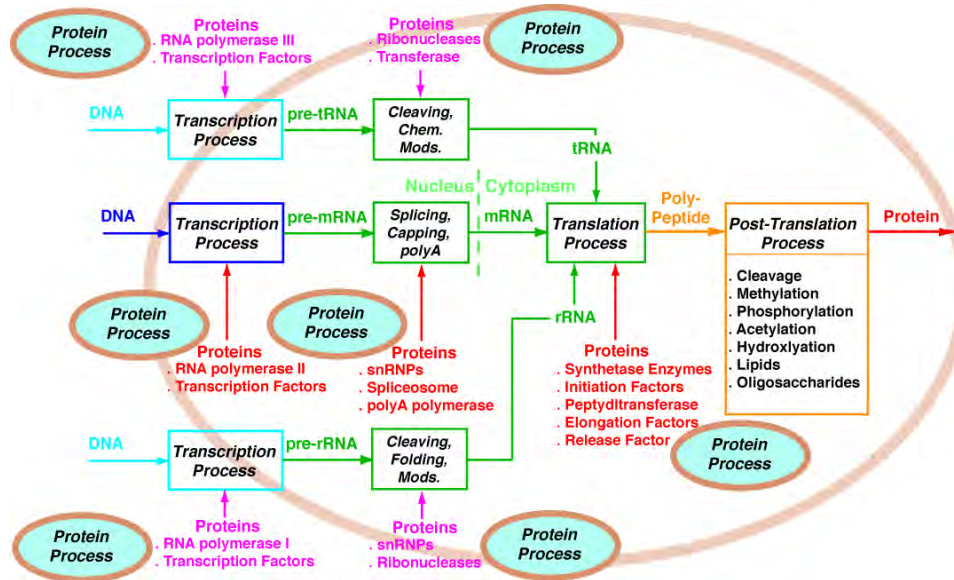


~20,000 human genes

~3 billion base pairs/human gamete

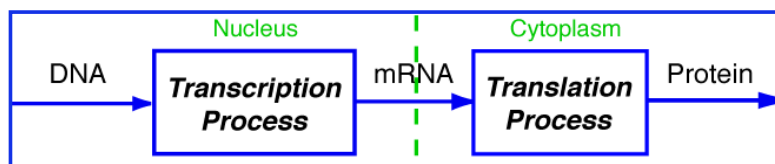
48

Block Diagram of the Protein Process



49

Protein Production is Dynamic



Transcription

Translation

Transcription

Duration: 1'13"
File Size: 5.2 MB
Contact: wehi-tv@wehi.edu.au

Translation

Duration: 2'27"
File Size: 11 MB
Contact: wehi-tv@wehi.edu.au

~50 bases/sec

~10 amino acids/sec

50

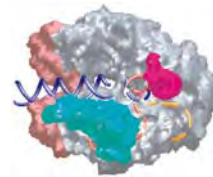
Polymerases and Ribosomes as Turing Machines



- Coding of amino acids from mRNA
- AUCG provides **Base 4 coding**

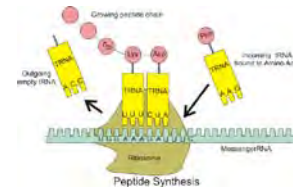
The Genetic Code				
First Position	Second Position	Third Position		
U	U	C	A	G
	F	S	Y	C
	L	S	Stop	Stop
C	L	S	Stop	W
	L	P	H	R
	L	P	Q	R
A	L	P	N	R
	I	T	N	S
	I	T	K	R
	I	T	D	R
G	V	A	K	G
	V	A	D	G
	V	A	E	G
	V	A	E	G

RNA Polymerase II*



* = Enzyme
= Catalytic protein

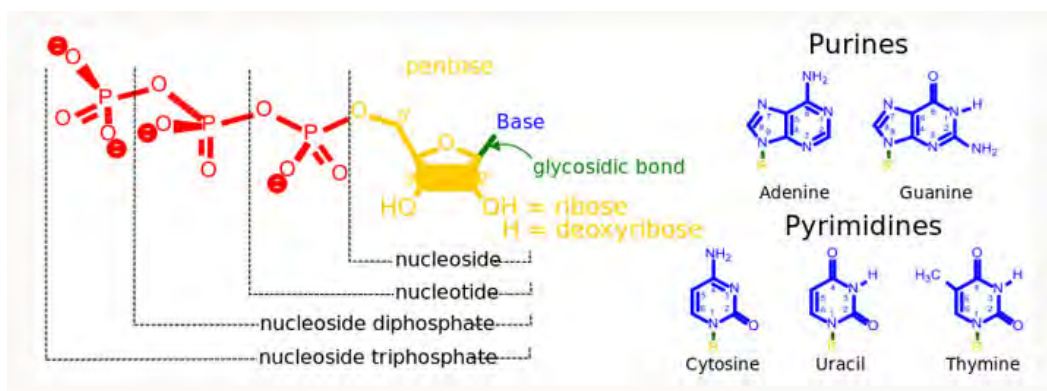
Ribosome**



** = Complex of
rRNA and
proteins

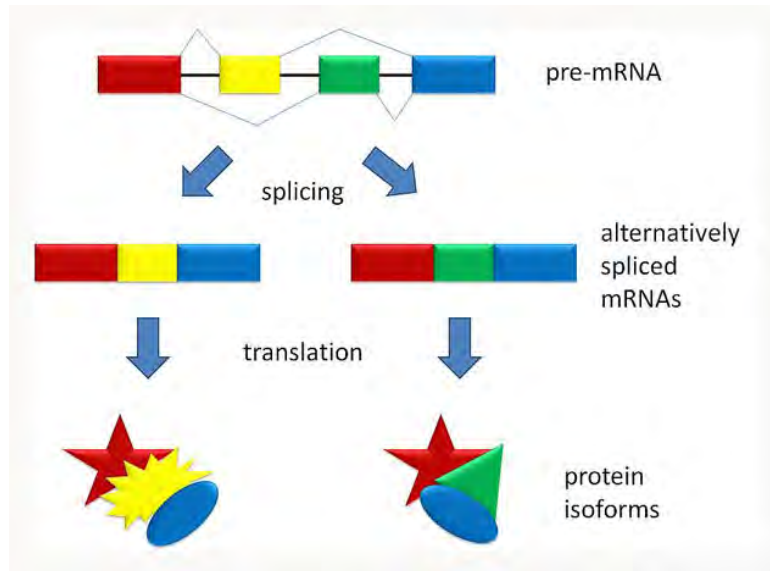
51

DNA/RNA Molecules

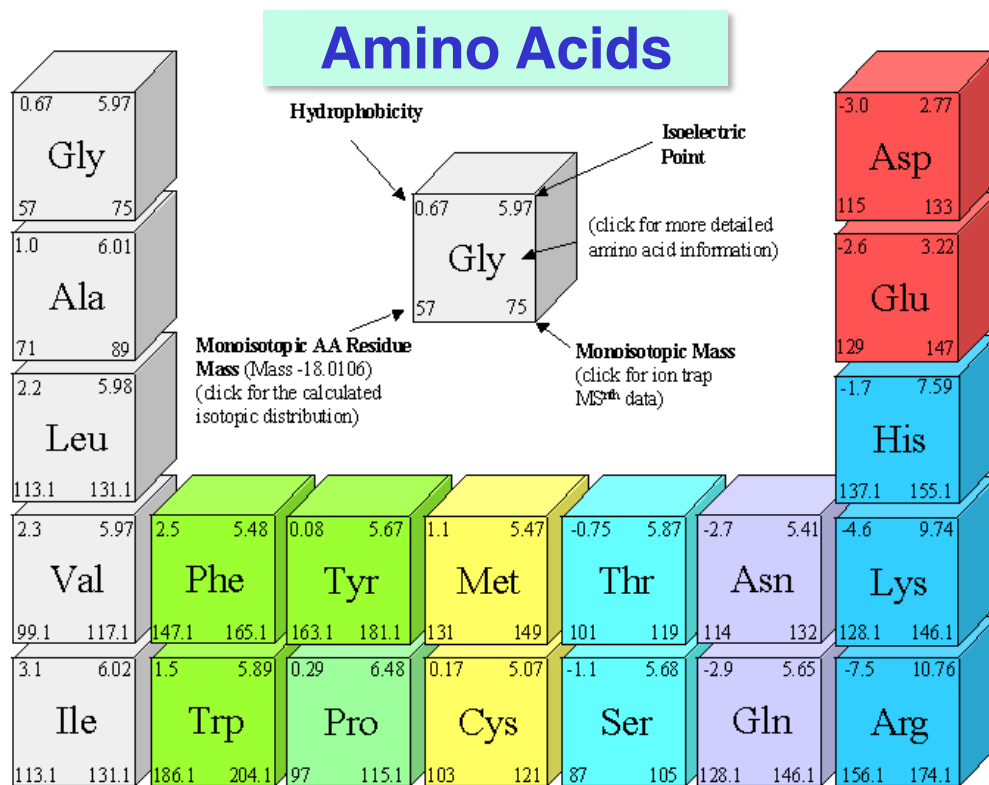


52

Alternative Splicing



53



54

How Do Slide Rules and Calculators Work?



- **Abacus**
 - **Unary number system**
 - <http://gwydir.demon.co.uk/jo/numbers/machine/abacus.htm>



- **Mathematical Tables**
 - **Manual calculations**
 - http://en.wikipedia.org/wiki/Mathematical_tables



- **Slide rule**
 - **Logarithmic scales**
 - <https://www.youtube.com/watch?v=uUzSSStVnAHk>

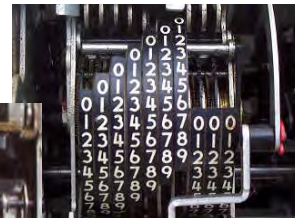


- **Mechanical Calculator**
 - **Add, subtract, and shift**
 - http://en.wikipedia.org/wiki/Mechanical_calculator
 - <https://www.youtube.com/watch?v=7S0BETniokI>

55



Friden STW-10 Calculator Mechanism



56

Bi-Quinary Control Panel Lights for the IBM 650



Computation with 10-digit words

57

Early Computers - DEC

PDP-1



<https://en.wikipedia.org/wiki/PDP-1>



SpaceWar



*PDP-6
Time-Sharing*



<https://en.wikipedia.org/wiki/PDP-6>

58



Microprocessor Transistor Counts 1971-2011 & Moore's Law



USAir Flight 427
Aliquippa, PA
September 8, 1994
Boeing 737-300

Probable Cause

- The *National Transportation Safety Board* determines that the **probable cause** of the *USAir flight 427* accident was
 - a **loss of control** of the airplane resulting from the movement of the rudder surface to its blowdown limit.
- The rudder surface **most likely deflected** in a direction **opposite to that commanded** by the pilots as a result of
 - a **jam** of the main rudder PCU servo valve secondary slide to the servo valve housing offset from its neutral position and
 - **overtravel** of the primary slide.