Formal Symbol Systems

"We obscure our knowledge with technical details in order to take a voyage of discovery that will eventually lead us back to what we already know."

-- Arthur "Arty" Fischell

Computing and Mathematics

-- David Parnas

Types of mathematical training needed by computer professionals:

calculus
discrete mathematics
logic
linear algebra
graph theory
differential equations
probability and applied statistics
optimization
numerical analysis

Software Engineering and Formal Methods

-- J. B. Wordsworth, in *Formal Methods of Software Engineering*

The roles in which software engineers need formal methods:

specification engineer
design engineer
programmer
documentation engineer
test engineer
service engineer

Approaches to Specification

-- Dean and Hinchey, in *Teaching and Learning Formal Methods*

"Natural language...is hopelessly inadequate when we have to deal unambiguously with situations of great intricacy...in such activities as legislation, arbitration, mathematics or programming."

Three Approaches

Only do what natural language can handle. *Exclude and redefine* intracacy.
Bend natural language to fit the purpose.
Design a completely new language (mathematics, programming).

Beauty

Scientific beauty consists of

1. simplicity (completeness, economy)
2. harmony (symmetry)
3. brilliance (clarity, connectedness)

"Beauty is the primary standard for scientific truth."
-- Augos and Staneiv, *The New Story of Science*, p.39

"You can recognize truth by its beauty and simplicity"
-- Richard Feynman

"Frequently a theorist will throw out a lot of data on the grounds that if they don't fit an elegant scheme, they are wrong."
-- Murray Gell-Mann

"A theory is more impressive the greater the simplicity of its premises is, the more different kinds of things it relates, and the more extended is its area of applicability."
-- Albret Einstein

The Modeling Hierarchy

Conceptualization	(imaginary, perceptual, ideal, cognitive)
Mathematical Model	(formal, symbolic, abstract, mathematical)
Data Structure and Algorithms	(representation, computational, software)
Machine Implementation	(actual, structured, physical hardware)

Formal Symbol Systems

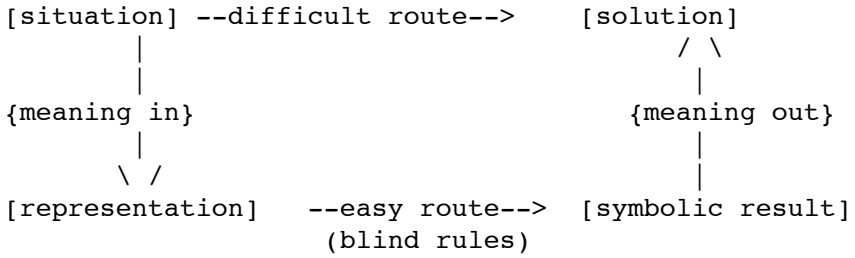
- a *universe* of discrete, stable, unique, disjoint, localized objects
- stable *maps* between objects. A territory with paths that can be traversed. A graph.
- *State Space*: the graph of states and transforms between states.
- an *interpretation* which maps symbols/tokens one-to-one onto objects/meanings.
- a defined and stable notion of *truth*.

Formal Modeling

A **formal system** consists of

- several sets of labels (for objects, functions, relations) called *constants*,
- rules for building *compound sentences* (or equations or expressions), and
- rules for *evaluating and simplifying* compound expressions.

Using a Formal System



Numeric or Symbolic Computation

Compute symbolically, unless no efficient symbolic technique is known;
then use optimized numeric techniques.

SYMBOLIC:
meaning
-- written as -->
symbol structures
-- reduced by -->
symbolic transformation rules
-- turning into -->
simpler symbol structures
-- read for -->
meaning

NUMERIC:
meaning
-- exemplified by -->
selected instances
-- substituted into -->
symbol structures
-- reduced by -->
numeric simplification rules
-- turning into -->
approximate results
-- read for -->
meaning

Computation

"A computational process is indeed much like a sorcerer's idea of spirit.
It cannot be seen or touched.
It is not composed of matter at all.
However, it is very real."
-- Abelson and Sussman,
Structure and Interpretation of Computer Programs, p.1

Computation is that which can be done by an *effective procedure*, or by a *Universal machine*.

EFFECTIVE PROCEDURE
well-founded recursive algorithm

UNIVERSAL MACHINE
Turing tape
programming language + instruction sequence
transistor network + timing
cellular automata
game of Life
FSM + stack
spreadsheet (like Excel)
database (like Access)

Models of computer

Levels of computational architecture

hierarchy of abstraction specification languages
machine language specification
vonNeumann tradeoff
circuit behavioral specification
hierarchy of realization specification languages

design model	abstract behavior
architecture model	abstract structure
performance model	abstract efficiency
correctness of behavior	functionality
efficiency of behavior	performance
actual behavior of physical circuit	reality

bit, word, instruction, program, message, application, user interface

Programming hierarchy

- *User interface*: metaphoric system which makes design interface accessible to non-experts.

- *Design interface*: hidden symbolic system which provide conceptual language for non-expert human to specify design abstractions.

- *Design abstraction*: pure symbolic system which expresses a human objective

- *High-level programming language*: symbolic system which closely models expert human models (math) and hides machine needs [Often math and algorithm are confused.]

- *Programming language*: symbolic system which expresses assembly steps in human writable form. Does not cleanly differentiate between requirements of the human and those of the machine.

- *Assembly language*: symbolic system which expresses machine language in process steps over specific logic function systems

- *Machine language*: symbolic system which transfers low level machine instructions into processes within designed physical logic function systems

- *Hardware design language:* symbolic system which specifies parallel operation of gate arrays

- *Logic function systems:* physical system, integrated networks of gates

- *Gates:* abstract physical system of logic operations and connecting wires

- *Transistors:* ignored physical system, assumed to be bundled in gates

Theories of Computation

All are formal, all are distinctly different.

- Formal Symbol Manipulation
  The result of a century of work in metamathematics.
  The mechanical manipulation of symbolic structures, without regard to meaning.

- Effective Computability
  What can be done mechanically by an abstract mathematical model of a machine.
  How difficult such a computation is to perform.

- Rule-base Reasoning, algorithm execution
  The behavior produced by following an explicit set of rules or transformations.
  How to construct rule following machines.

- Digital State Machines
  Circuits and machines with a finite, disjoint set of internal, homogeneous states.
  A state is an array of values or configurations when time is stopped.

- Information Processing
  Storing, manipulating, displaying, and transferring "information"
  Encoding utility in vast arrays of bits.

- Physical Symbol Systems
  Computers are made of, and interact with, symbols, in a way which depends upon

the physical embodiment of the symbols.
Physical embodiment occurs in bit arrays and streams, in instruction sets, in cellular arrays, and in minds.

- Interactive Agents
  Agents embodied in an environment interact and communicate with each other.
  Atomic programs with specific functionality moving in an emergent network.

- Non-linear Dynamic Systems
  Fractals, chaos theory.
  Non-linear phenomena and computations,
  modelled by solutions to differential equations and iterative systems.

- Complex Adaptive Systems
  Artificial life, complex systems.
  Entities and programs in array-based environments which respond to changes in the environment by survival adjustments.
  Learning through evolution

- Clockwork Universe
  Dated: 17th and 18th centuries.
  The Universe and everything in it
  is like the mechanism of a large, perfectly tuned clock.

- Quantum Computing
  All observation and behavior is generated by the "collapse"
  of a superposed wave function.
  Probabilistic indistinguishability yields reality through measurement.

- Molecular Computing
  Massive numbers (~10^23) of molecules transact chemical and physical exchanges.
  Results are the statistical average of dominant configurations (i.e. types of molecules)

- Biological Computing
  DNA and RNA structures perform replicate pattern-matching.
  Results are functional systems whcih compute behavior in an environment.

- Agoric Computing
  Economic agents and programs transact tokens of value to accumulate processing resources.
  Competitive survival of function in a free market.

The Virtues of Mathematical Models

-- Gries and Schneider, *A Logical Approach to Discrete Math*

- A mathematical model may be more *understandable, concise, precise, or rigorous*

than an informal description in natural language.

- Answers to questions about an object or phenomenon can often be *computed directly* using a mathematical model of the object or phenomenon.

- Mathematics provides *methods for reasoning*:
  for *manipulating* expressions,
  for *proving* properties, and
  for *obtaining new results* from known facts.
  This reasoning can be done without knowing or caring what the symbols being manipulated mean.

The Trouble with Mathematical Models

- Only a *small portion of the world* and of our experience can be discretely objectified.

- *Abstraction discards information*.

- Modeling does not reflect *human processes*.
  (Students taught how to think in models generally make poor programmers.)

- Modeling dictates a worldview which, at times, *may be dysfunctional*.
  "Present mathematical and scientific education is a hotbed of authoritarianism and is the worst enemy of independent and critical thought." -- Lakatos

- Human reasoning is *physiologically mediated* by human emotion.
  Discovery is intuitive and involves guesswork.

The Pattern of Growth of Theories

-- Lakatos, *Proofs and Refutations*

1. *Primitive conjecture*
2. *Proof* (a rough thought experiment)
3. *Global counterexamples* emerge, questioning the proof.
4. *Proof reexamined*. The incorrect portion of the proof is made explicit, and either the conjecture is limited or the definitions are broadened.

Steps 1-4 are the basic cycle, also

5. The *proofs of other theorems are examined* to see if the newly corrected portion (from 4) is relevant.
6. *Consequences* of the proof are examined.
7. *Counterexamples* are turned into new examples, and new theories or fields are created.