Introduction: Miser Project conception starts from reasoning with and in certain mathematical structures. That foundation is used with mathematical thinking in establishing a **model of computation** and subsequent **reasoning about programs** and **their dependable employment for practical purposes**.

The Miser Project use of "model" is reserved for a mathematical- logic situation around interpretations of theories, perhaps a [third flavor](#) of modeling (or theory-building) with respect to **the marvelous utility of computers as instruments of human purpose and experienc**e.

* Mind-map notes and comments by William L. Anderson,
https://orcid.org/0000-0003-3200-7947, 2019,
CC0 https://creativecommons..org/publicdomain/zero/1.0

6. Manifestation and Tacit Awareness

5. Choreography of Computation

4+. Concerning Representational Equivalence

4. Interpretation: Variations on a Theme

[Miser Project: Interpretation, Representation, Computation, and Manifestation](#)*
Dennis E. Hamilton (Posted on 2019-02-11-14:24)

1. Recap: Computation-Model Stage-Setting†

† I need definitions (examples?) for:
- model of computation
- effective-computability[1]
- operational computational process
- computational representation
- mathematically-demonstrated correspondence

A. With regard to computational interpretation, effective-computability is taken as a condition on deductions that are available given a particular mathematical representation of a function or predicate.

B. The transition from an effectively-computable representation in any structure(?) to an operational computational process of any kind is suggested, not demonstrated.

C. It is posited that the interpretation-in-‹ob› approach is sufficient for computational representation of the same effectively-computable mathematical functions as any other recognized model of computation. Demonstration of such universality (i.e., [Church-Turing computability](#)) and common limitations of that universality is **one Miser Project objective:** making this aspect of theoretical computing accessible in practical, demonstrable, and understandable terms for computing practitioners and other interested parties.

2. Structure, Interpretation, Representation, Computation, and Manifestation†
Some informal notions are adapted and restricted in the context of the Miser Project to assist in capturing the essence of computation and its practical application.

Structure: mathematical characterization of theoretical entities, in terms of domains, primitive notions, and applied logical theories.

Interpretation:
- *mathematical:* arrangement by which one mathematical structure is modeled in another using a mathematically-demonstrated correspondence(?)
- *linguistic*: interpretation of a text in a (standard) semantic domain
- *empirical*: achievement of engineered mechanized operations confirmed to satisfy a model of computation
- *social / scientific:* identifying objects in physical/social reality as evocative of theoretical entities, and sometimes *vice versa*

Representation:
- *mathematical*: formulation of mathematical functions in a given structure's theory language
- *conceptual*
- *reflective*

Computation:
- *conceptual*: performance of (algorithmic) procedure p such that representation of some entity x, the operand/data, is transformed into a representation of y, satisfying specified functional relationship y = f(x).
- *informal*: the result of a computation, e.g., "the square root of x"

Manifestation:
- *general*
- *operational*
- *interactive*
- *tacit*

3. Review: More About Structure

References:
1. M. Davis, *Computability and Unsolvability*, Dover Publications, NY, NY, 1982, ISBN 0-486-61471-9, pp. xv-xviii.