Abstraction in Education

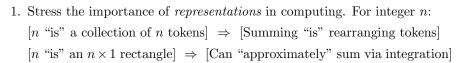
Arnold L. Rosenberg Distinguished University Professor Emeritus University of Massachusetts

It is said that if you give a person a fish, you feed him/her for a day. If you teach the person how to fish, you feed him/her for a lifetime.

This aphorism has been my roadmap, as a learner and a teacher. Inspired by the aphorism, I strive to:

- educate rather than train
- abstract from concrete and immediate to fundamental and essential
- * Inspire each student to become a mathematician or an informaticist, rather than just *learn* some mathematics or some informatics.
- * Equipped with understanding of "why?" rather than just "how?" a student can confront new situations with courage and creativity.

Examples:



- 2. Study parallel computing via dependency dags, not programs — identify sources of concurrency.
 - (a) Expansion-Reduction computations:
 - Shared structure: $\begin{cases} \bullet & Parallel \ mergesort \\ \bullet & Numerical \ integration \end{cases}$

(b) Parallel-Prefix computations: Shared structure: $\begin{cases} \bullet & Carry-lookahead \ addition \\ \bullet & Computing \ paths \ in \ graphs \end{cases}$

(c) Butterfly-Structured computations: Shared structure: $\begin{cases} \bullet & Convolutions \ (e.g., \ polynomial \ multiplication) \\ \bullet & Odd-even \ mergesort \end{cases}$

3. Understand the logical complexity of computation —e.g., self-referential systems