**Infinite Computation, Co-induction and Computational Logic**

Coinduction is the dual of induction. Induction corresponds to well-founded structures that start from a basis which serves as the foundation: e.g., natural numbers are inductively defined via the base element zero and the successor function. Inductive definitions have 3 components: initiality, iteration and minimality. For example, the inductive definition of lists of numbers is as follows: (i) [] (empty list) is a list (initiality); (ii) [H|T] is a a list if T is a list and H is some number (iteration); and, (iii) the set of lists is the smallest set satisfying (i) and (ii) (minimality). Minimality implies that infinite-length lists of numbers are not members of the inductively defined set of lists of numbers. Inductive definitions correspond to least fixed point (LFP) interpretations of recursive definitions. Coinduction eliminates the initiality condition and replaces the minimality condition with maximality. Thus, the coinductive definition of a list of numbers is: (i) [H|T] is as a list if T is a list and H is some number (iteration); and, (ii) the set of lists is the largest set of lists satisfying (i) (maximality). There is no base case in a coinductive definition, and while it may appear circular, the definition is well formed since coinduction corresponds to the greatest fixed point (GFP) interpretation of recursive definitions. Thus, the set of lists under coinduction is the set of all infinite lists of numbers (no finite lists are contained in this set). Note, however, that if we have a recursive definition with a base case, then under the coinductive interpretation, the set defined will contain both finite and infinite-sized elements, since in this case the GFP will also contain the LFP. A coinductive proof essentially is an infinite-length proof.