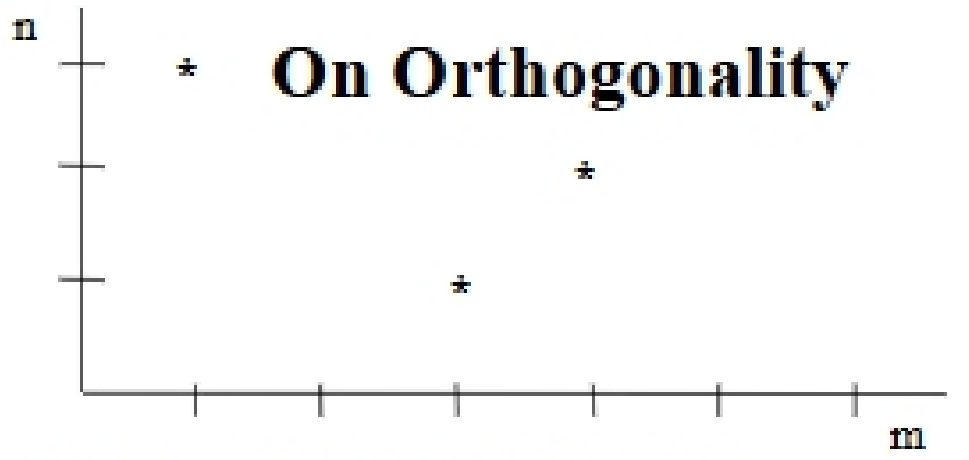

On Orthogonality

- What makes orthogonality work?
  - by remembering only m + n things, we get m * n capabilities.
- Orthogonality says that no point should be definable by more than one XY pair.
- Orthogonality advantageous only if:

$$m+n+e < m*n-e$$

ALGOL68

ALGOL68: Goals & History

Thesis: *It is good practice in programming language design to abstain from exceptions.*

- Design goals:
  - gen purpose, rigorously-defined language
  - Clear up trouble spots in ALGOL60
    - (but, Pascal more like A60 than A68 is)
  - orthogonality, extensibility
- ALGOL68 - development started in mid-60's.
  - Revised report (SIGPLAN Notices, May 1977) cleared up many ambiguities.

Key Ideas in ALGOL68

- User type declarations (modes)
- Orthogonal design (modes, structures, ops)
- Reference mode (pointers of a sort)
- United modes (predecessor to variant records)
- Auto declaration of FOR LOOP index
- User-specified operator overloading
- Notion of "elaboration" on context entry

More Key Ideas

- Mode requirement for formals
- Casting: user-spec'd mode conversion
- Redefinition of operator precedence
- Collateral actions
- Semaphores
- W-grammars - two-level grammar
- Contexts (strong, firm, meek, weak, soft)
  - WRT coercion

ALGOL68 Structure

- ALGOL68 is block structured w/ static scope rules
  - Monolithic programming, as in ALGOL60 (and later in Pascal)
- ALGOL68's model of computation:
  - static
  - stack: block/procedure AR's; local data objects
  - heap: "heap" -- dynamic-- data objects
- ALGOL68 is an *expression*-oriented language
  - (note influence on C/C++)