Derived Variables

A derived variable, \( v \), is a function assigning a “value” to each state:

\[
v: \text{States} \rightarrow \text{Values}
\]

If \( \text{Vals} = \mathbb{N} \), say \( v \) is “\( \mathbb{N} \)-valued” or “nonnegative-integer-valued”

Robot on the grid example:

States = \( \mathbb{N}^2 \). Define the sum-value, \( \sigma \), of a state:

\[
\sigma(x,y) ::= x+y
\]

an \( \mathbb{N} \)-valued derived variable

Called derived to distinguish from actual variables that appear in a program.

For robot

Actual: \( x, y \)

Derived: \( \sigma \)
Another derived variable:
\[ \pi ::= \sigma \mod 2 \]
\( \pi \) is \( \{0,1\} \)-valued

For Fast Exp, have (actual) variable \( Z \).
Proof of termination:
\( Z \) is strictly decreasing & natural number-valued

Termination followed by
Well Ordering Principle:
\( Z \) must take a least value.
then the algorithm is stuck

Goes down at every step
Weakly Decreasing Variable

Down or constant after each step

Diagonal Robot variables

σ: up & down all over the place
neither increasing
nor decreasing

π: is constant
both weakly increasing
& weakly decreasing

(We used to call weakly decreasing variables "nonincreasing" variables.)

OK terminology but remember:
nonincreasing is NOT SAME as "not increasing:"

...
Well ordered sets

Def. A set W of real numbers is well ordered iff it has no infinite decreasing sequence
\[ w_0 > w_1 > w_2 > \ldots > w_n > \ldots \]

Termination using WOP on \( \mathbb{N} \) generalizes to strictly decreasing variables whose values are in any well ordered set.