Recursive Algorithms Illustrating Proofs by Induction

E. L. Lady

Theorem. A set consisting of n elements has exactly 2^n subsets.

The proof of this theorem by induction is illustrated by the following recursive algorithm for listing all the subsets of a set.

procedure `list-subsets(`$A_n = \{x_1, \ldots, x_n\}$`:` set with exactly `n` elements)
  if `n = 0` **then** `list ∅`
  else **begin**
    `list-subsets(`A_{n-1}`)`
    `Adjoin` x_n `to each subset listed in the line above and list the resulting sets.`
    end

{ All the subsets of A_n have been listed.
 The subsets of A_{n-1} are the same as the subsets of A_n not containing x_n.
 Each subset of A_{n-1} produces exactly one subset of A_n containing x_n.
 Thus A_n has twice as many subsets as A_{n-1}. }

Theorem. A $2^n \times 2^n$ checkerboard with one square deleted can be tiled by an L-shaped tile three squares large.

The proof of this theorem by induction is illustrated by the following recursive algorithm for tiling the checkerboard.

procedure `tile(`C_n`:` a $2^n \times 2^n$ checkerboard with one square deleted)
  if `n = 1` **then** `lay down a single tile in the only possible way`
  else **begin**
    `Cut the board up into four` $2^{n-1} \times 2^{n-1}$ `sub-boards`
    `tile(the sub-board with the missing piece)`
    for `i = 1 to 3`
      begin
      `B:= the next sub-board not already tiled`
      `B:= B with the central corner removed`
      `tile(B)`
      end
    `tile(the three squares in the middle that are still untiled)`
    end