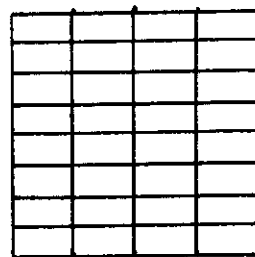# Nondeterministic construction of an arbitrary witness.

A series of problems that is well-known in educational circles starts with the following one: "Does the 8-by-8 square admit a tiling with 32 dominoes of 2-by-1 ?". The answer is "Yes", the existence of such a tiling is traditionally demonstrated by displaying what is called "a witness", i.e. a way of tiling the square with the dominoes. The most common witness is: The argument that it is possible to tile the square is utterly convincing, and traditionally one immediately moves to the next problem in the series.

But, short and convincing as the above argument may be, it has what can be viewed as a shortcoming: it is grossly overspecific! The 8-by-8 square can be tiled in thousands and thousands of ways and there is no way of justifying the specific choice made above. This consideration raises the following question: can we design a program that (i) provably constructs a tiling of the square, and (ii) is so non-

deterministic, that any tiling may be con-
structed?

*   *   *

In order to reduce the number of ways
in which our program may generate a
specific tiling, we rank the dominoes of
a tiling with natural numbers such that
for any pair of neighbours the southern
or the western one has the lower rank,
for instance for a 4-by-4 square
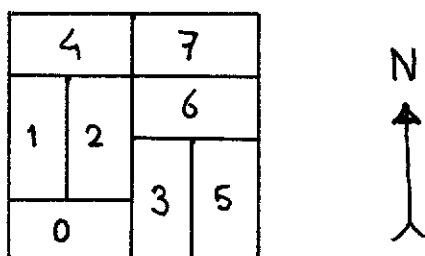
(Note that the tiling need not determine the
ranking uniquely: in the above example,
ranks 3 and 4 could have been inter-
changed, and so could ranks 4 and 5.)

Our program will "place" the dominoes
of the tiling it generates in the order of
increasing rank, but in order to be sure
that it can generate any tiling, we have
to convince ourselves of

Lemma 0   Each tiling admits a ranking
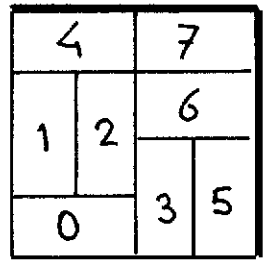of its dominoes.

2

Proof Consider the path from the SE corner to the NW corner with precisely the dominoes with rank $< n$ to its left. In the above example we show the cases $n = 0$, $n = 4$, and $n = 8$:
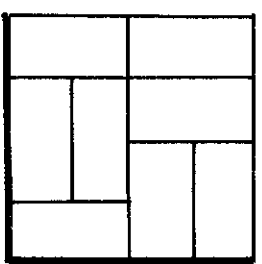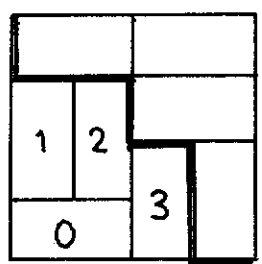
n=0   n=4   n=8

It suffices to show that for any specific tiling we can assign the ranks in increasing order. Consider now the paths with n equal to the number of dominoes ranked. Note that all these paths have the same length: by the definition of ranking, the "vertical" segments are traversed towards the North, the horizontal ones from East to West.

Assigning the next rank, including the increase of n by 1 and the adjustment of the path, is by ranking a "horizontal" or "vertical" domino:

horizontal: becomes

vertical: becomes

3
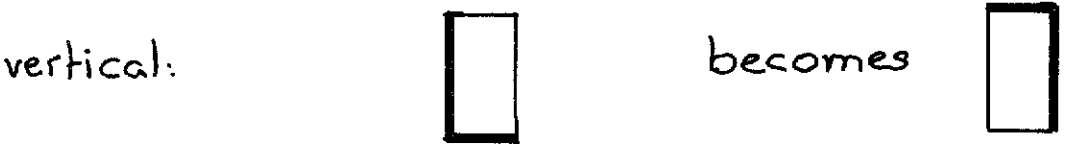
Such a move is possible provided there exists an unranked domino with its West side and its South side entirely along the path. We have to show the existence of such a domino when not all dominoes have been ranked.

When not all dominoes have been ranked, the path contains at least 1 right turn:

These turns come in two flavours, depending on the orientation of the adjacent unranked domino:
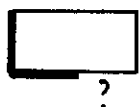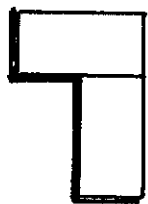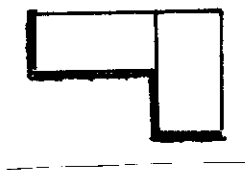
H-turn: V-turn: ?

If on the path from the SE corner to the NW corner, the first right turn is an H-turn, the adjacent horizontal domino can be ranked; if the last right turn is a V-turn, the adjacent vertical domino can be ranked. If neither is the case, the path contains a V-turn followed by an H-turn:

,   , or further apart. Always at least one of the adjacent dominoes can be ranked. (End of Proof.)

4

The proof of Lemma 0 was the hard part be-
cause I needed -at least: used- all those pic-
tures; now comes the simple part, surprisingly
simple even. Thanks to Lemma 0 , and in
terms of the concepts introduced in its
proof, our task is now to design a program
that transforms, in any possible way , but
without backtracking, in 32 instances of
the ranking step the path for n=0 into the
path corresponding to n= 32 .

I propose to characterize the shape of
the path by listing in succession , for in-
stance in the order from the NW corner
to the SE corner the direction of the 16
edges along the path, with s (for south)
and e (for east) . The initial path is

s s s s s s s s e e e e e e e e   ,

the final path is

e e e e e e ee ss ss ss ss   .

The two ranking steps operate — see the
bottom of EWD1229-2 — on three succes-
sive elements of this sequence:

horizontal: .... s e e.... becomes ...e e s ....

vertical: ... s s e.... becomes .... e ss ....

But these two steps are both of the form

.... s?e....   becomes ....e?s....   !

Any pair that is one apart and "out of order" may be interchanged. We see an arbitrary interleaving of two copies —one applied to the elements in the even positions and one applied to the elements in the odd positions— of the nondeterministic algorithm that transforms ssseeee into eeeessss by 16 swaps of neighbours, each replacing ...se... by ...es... . Our program termin- ates without backtracking in 32 swaps.

Austin, 21 January 1996

prof. dr. Edsger W. Dijkstra
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-1188
USA