

Why "heavy-weight" bits are unavoidable.

The following statement from my "On the cruelty of really teaching computing science" was recently challenged in an as yet private communication:

"In the discrete world of computing, there is no meaningful metric in which "small" changes and "small" effects go hand in hand, and there never will be."

Here is the type of reasoning that led to the above conclusion.

We restrict ourselves for simplicity's sake to a store of n bits, for which we consider the inversion of a single bit as the smallest possible change. With each storage configuration we associate a value, and on the corresponding value space we envisage a metric that defines a distance $d(x,y)$ for any pair of values (x,y) . Besides the obvious $d(x,x) = 0$ and $d(x,y) = d(y,x)$, the metric being "meaningful" is meant to imply the validity of the triangular inequality, viz. $d(x,y) \leq d(x,z) + d(z,y)$.

Assume now that our metric is such that the distance between two values that correspond to storage configurations that differ in precisely one bit is (at most) 1. (This assumption is intended to model that "small" changes and "small" effects go hand in hand.) From the fact that the store has n bits and the triangular inequality we now deduce that the distance between any two values that correspond to storage configurations is at most n . But this upper bound is much too small to be economically acceptable: instead of storing in n bits one of the values from 0 through n , we know how to store in those n bits one of the values from 0 through $2^n - 1$. So, in my quoted statement "never" means "not as long as we refuse to pay an exponential price while a linear one is feasible".

Austin, 30 April 1992

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