The Infinity Of Lists

Umberto Eco

Programming with infinite lists open up a new way of writing programs, where you give rules for generating a sequence rather than building it by hand often leading to simpler code. Since data types are lazy and support self references, you can produce a list dynamically by defining it as a function of its successors. The simplest example of this is the sequence of natural numbers (ie. \(1,2,3,4,\ldots\)) there are a few different ways of doing this in Haskell:

```
numbers1 = 1 : map (+1) numbers1
numbers2 = 1 : [x+1 | x take 10 numbers1]
```

```
Main> take 10 numbers1 [1,2,3,4,5,6,7,8,9,10]
Main> take 10 numbers2 [1,2,3,4,5,6, \#
```

*...a splendidly illustrated monograph, The Infinity of Lists: An Illustrated Essay (Rizzoli) is, in essence, a tour through art, literature, and music based on the theme of lists, an investigation of the phenomenon of cataloging and collecting. Additionally, Eco maintains that the impulse to accumulate, to collect, is a reoccurring passion in Western culture."*Â The chief theme is the attempt by authors and artists to express comprehensiveness, incomprehensibility, and even infinity with lists (whether verbal or artistic). Sometimes the book drags, and sometimes the extracts are just too long. Or boring.