1 Introduction

Much of computer science is about state machines. This is as obvious a remark as saying that much of physics is about equations. Why state something so obvious?

Imagine a world in which physicists did not have a single concept of equations or a standard notation for writing them. Suppose that physicists studying relativity wrote the "einsteinian" $m \nearrow c^2 \hookleftarrow E$ instead of $E = mc^2$, while those studying quantum mechanics wrote the "heisenbergian" $\underset{\wedge}{\underline{E \quad \overset{c^2}{m}}}$; and that physicists were so focused on the syntax that few realized that these were two ways of writing the same thing. In such a world, it would be worth observing that relativity and quantum mechanics both used equations.

This imagined world of physics seems absurd. Its analog is the reality of computer science today. Computation is a major topic of computer science, and almost every object that computes is naturally viewed as a state machine. Yet computer scientists are so focused on the languages used to describe computation that they are largely unaware that those languages are all describing state machines.

Teaching our imaginary physicists that einsteinians and heisenbergians are different ways of writing equations would not lead to any new physics. The equations of relativity are different from those of quantum mechanics. Similarly, realizing that so much of computer science is about state machines might not change the daily life of a computer scientist. The state machines that arise in different fields of computer science differ in important ways, and they may be best described with different languages. Still, it seems worthwhile to point out what they have in common.

State machines provide a framework for much of computer science. They can be described and manipulated with ordinary, everyday mathematics—that is, with sets, functions, and simple logic. State machines therefore provide a uniform way to describe computation with simple mathematics.

The obsession with language is a strong obstacle to any attempt at unifying different parts of computer science. When one thinks only in terms of language, linguistic differences obscure fundamental similarities. Simple ideas can become complicated when they must be expressed in a particular language. A recurring theme is the difficulty that arises when necessary concepts cannot be introduced either because the language has no way of expressing them or because they are considered to be politically incorrect. (A number of different terms have been used to mean politically correct, such as "fully abstract", "observable", and "denotational".)

Here is a simple example. Suppose we have a chain complex

$$\cdots \xrightarrow{\partial} C_{d+1} \xrightarrow{\partial} C_d \xrightarrow{\partial} C_{d-1} \xrightarrow{\partial} \cdots$$

that is "graded", i.e., each C_d splits into a direct sum

$$C_d = \bigoplus_{p=1}^{n} C_{d,p}$$

and moreover the boundary map ∂ respects the grading in the sense that $\partial C_{d,p} \subseteq C_{d-1,p}$ for all d and p. Then the grading allows us to break up the computation of the homology into smaller pieces: simply compute the homology in each grade independently and then sum them all up to obtain the homology of the original complex.

Unfortunately, in practice we are not always so lucky as to have a grading on our complex. What we frequently have instead is a *filtered complex,* i.e., each C_d comes equipped with a nested sequence of submodules

$$0 = C_{d,0} \subseteq C_{d,1} \subseteq C_{d,2} \subseteq \cdots \subseteq C_{d,n} = C_d$$

and the boundary map respects the filtration in the sense that

$$\partial C_{d,p} \subseteq C_{d-1,p} \tag{1}$$

for all d and p. (Note: The index p is called the *filtration degree.* Here it has a natural meaning only if $0 \le p \le n$, but throughout this paper, we sometimes allow indices to "go out of bounds," with the understanding that the objects in question are zero in that case. For example, $C_{d,-1} = 0$.)

Although a filtered complex is not quite the same as a graded complex, it is similar enough that we might wonder if a similar "divide and conquer" strategy works here. For example, is there a natural way to break up the homology groups of a filtered complex into a direct sum? The answer turns out to be yes, but the situation is surprisingly complicated. As we shall now see, the analysis leads directly to the concept of a spectral sequence.

Let us begin by trying naïvely to "reduce" this problem to the previously solved problem of graded complexes. To do this we need to express each C_d as a direct sum. Now, C_d is certainly not a direct sum of the $C_{d,p}$; indeed, $C_{d,n}$ is already all of C_d. However, because C_d is a finite-dimensional vector space (recall the assumptions we made at the outset), we can obtain a space isomorphic to C_d by modding out by any subspace U and then direct summing with U; that is to say, $C_d \simeq (C_d/U) \oplus U$. In particular, we can take $U = C_{d,n-1}$. Then we can iterate this process to break U itself down into a direct sum, and continue all the way down. More formally, define

$$E^0_{d,p} \stackrel{\text{def}}{=} C_{d,p}/C_{d,p-1} \tag{2}$$

for all d and p. (*Warning:* There exist different indexing conventions for spectral sequences; most authors write $E^0_{p,q}$ where $q = d - p$ is called the *complementary degree.* The indexing convention I use here is the one that I feel is clearest pedagogically.) Then

$$C_d \simeq \bigoplus_{p=1}^{n} E^0_{d,p}. \tag{3}$$

The nice thing about this direct sum decomposition is that the boundary map ∂ naturally induces a map

$$\partial^0 : \bigoplus_{p=1}^{n} E^0_{d,p} \to \bigoplus_{p=1}^{n} E^0_{d-1,p}$$

such that $\partial^0 E^0_{d,p} \subseteq E^0_{d-1,p}$ for all d and p. The reason is that two elements of $C_{d,p}$ that differ by an element of $C_{d,p-1}$ get mapped to elements of $C_{d-1,p}$ that differ by an element of $\partial C_{d,p-1} \subseteq C_{d-1,p-1}$, by equation (1).

Therefore we obtain a graded complex that splits up into n pieces:

$$\begin{array}{ccccccccc}
\cdots & \xrightarrow{\partial^0} & E^0_{d+1,n} & \xrightarrow{\partial^0} & E^0_{d,n} & \xrightarrow{\partial^0} & E^0_{d-1,n} & \xrightarrow{\partial^0} & \cdots \\
\cdots & \xrightarrow{\partial^0} & E^0_{d+1,n-1} & \xrightarrow{\partial^0} & E^0_{d,n-1} & \xrightarrow{\partial^0} & E^0_{d-1,n-1} & \xrightarrow{\partial^0} & \cdots \\
 & & \vdots & & \vdots & & \vdots & & \\
\cdots & \xrightarrow{\partial^0} & E^0_{d+1,1} & \xrightarrow{\partial^0} & E^0_{d,1} & \xrightarrow{\partial^0} & E^0_{d-1,1} & \xrightarrow{\partial^0} & \cdots
\end{array} \tag{4}$$

Now let us define $E^1_{d,p}$ to be the pth graded piece of the homology of this complex:

$$E^1_{d,p} \stackrel{\text{def}}{=} H_d(E^0_{d,p}) = \frac{\ker \partial^0 : E^0_{d,p} \to E^0_{d-1,p}}{\operatorname{im} \partial^0 : E^0_{d+1,p} \to E^0_{d,p}} \tag{5}$$

(For those comfortable with relative homology, note that $E^1_{d,p}$ is just the relative homology group $H_d(C_p, C_{p-1})$.) Still thinking naïvely, we might hope that

$$\bigoplus_{p=1}^{n} E^1_{d,p} \tag{6}$$

is the homology of our original complex. Unfortunately, this is too simple to be true. Although *each term* in the the complex $(\bigoplus_p E^0_{d,p}, \partial^0)$—known as the *associated graded complex* of our original filtered complex (C_d, ∂)—is isomorphic to the corresponding term in our original complex, this does not guarantee that the two complexes will be isomorphic *as chain complexes.* So although $\bigoplus_p E^1_{d,p}$ does indeed give the homology of the associated graded complex, it may *not* give the homology of the original complex.