While the first two chapters covered fundamental ideas underlying computer hardware, in chapter 3 we jump all the way up to the level of software, with an overview of the act of *programming* a computer. The author compares programming languages and their constructs to those of natural human languages, making the crucial distinction that programming languages are *formal languages*, with strict syntactic and semantic rules, contrasting the ambiguity inherent in how humans parse spoken language. We get an introduction to functional abstraction via the toy programming language Logo, as well as the treatment of program instructions as data. We finish with the briefest of introductions to the idea of program translation, and a high-level overview of the complete process by which a computer program is written, translated, and executed.

Chapter 4 is mostly an introduction to concepts in computability theory. We start with a brief explanation of Turing machines, the main idea of the section being that all computing devices are essentially equivalent; they are all able to perform the exact same set of computations, regardless of differential speeds or memory sizes. There's a section about RNG, which just explains the fundamentals (chaotic systems, true randomness, etc.), followed by a section on about *noncomputable* problems and the Church thesis. We finish with some ideas about quantum computing that are, from what I understand, quite outdated, though theoretically sound.

I'm certainly not surprised by anything in chapter 3, but that doesn't mean it isn't interesting. I'm impressed by how concisely Hillis is able to explain all the most foundational ideas of programming. Logo seems like an excellent language to teach these ideas to children, and the progression presented by Hillis is logical and self-justifying. In only a few pages we cover subroutines, functional abstraction, loops, conditionals, and even recursion. As someone interested in education and pedagogical research, I enjoy reading about such incredibly well-constructed pathways for teaching complex ideas to children, some of which I didn't encounter until my second semester of university.

Chapter 4 is also interesting. I've not taken the Theory of Computation course here at Mines yet, so my exposure to these concepts in detail has been so far minimal. The idea that the human brain could, in principle, be simulated by a computer has come up before for me, but I've never considered the implication that this means the human brain is really just a Turing machine, no different from any other computational device. While the thought is, as Hillis suggests, rather troubling at first, I don't know if it really ought to be. Why should the human brain be any different from a computer? I think I agree with Hillis that, if this is indeed the case, it doesn't diminish the value of human thought and experience, only gives us a new way to think about and explore it's boundaries. Finally, the other idea I found interesting from this chapter was about random numbers. I had never really thought about the fact that you could generate a truly random sequence of data from something as simple as a Geiger counter, due to the laws of quantum mechanics. I'll need to factor that into my own theories about determinism and free will, just as soon as catch up on all this homework...