Review of David Harel, Computer Ltd.

We are used to expect better and more from our computers, but for a very large number of tasks no amount of money, time or ingenuity will ever be sufficient to build the right machine. Consider the history of flight. Given our present understanding of physics, we know that nothing can fly faster than light. Computers have comparable constraints. The topic is fascinating and full of important consequences. It is also technically complex, but David Harel has provided an excellent introduction.

Imagine an ideal computer within a prison. The first wall is the conceptually Incomputable. A robot cannot tide up a messy room or learn how to translate Homer because these tasks cannot be transformed into algorithms, the only sort of logical recipe that a computer can follow. Digital agents cannot acquire experience and common-sense, and this is why any strong programme in Artificial Intelligence is doomed to fail. The second wall is the logically Undecidable. A few mathematical theorems prove that a number of computational problems have no solution, regardless of the resources available to any ideal computer. Remember the Y2K problem? It was caused by human shortsight, and it had to be solved by human efforts. It is impossible to build a universal software verifier, which can automatically and fully evaluate whether a given program solves a given algorithm problem, for any program and algorithm we wish to test. Programming and testing software for correctness are empirical tasks that have to be left to engineers. The third wall is the physically Intractable. Some algorithmic problems are computable only in theory. In practice, they would require more time than the life of the universe, and more recording space than all its particles. Chess is, computationally speaking, intractable. Deep Blue does not play like a grand master. It can win against the world chess champion because it relies on rules of thumb and million of games and matches studied and played by generations of players. Playing against it is a bit like playing trivial pursuit against the *Encyclopaedia Britannica*. The fourth wall is not discussed in Harel's book, but it is equally constraining. It is the economically Unaffordable. Civilian airplanes could easily fly at the speed of sound but they don't for economic reasons, and the Concorde remains an exception. The same logic governs the development of ICT (information and communication technology). For example, chips made using gallium arsenide are much faster than silicon chips, but they are too expensive to manufacture and too dangerous to handle (arsenic is very toxic), hence their use is limited to special applications. Some computational tasks will remain forever too expensive, and not just for the general user.

As Harel shows in the second part of the book, the Undecidable is a logical wall that will never fall or move, and ingenuity aided by parallel, quantum, probabilistic or molecular computing could only help to shift the other three walls an inch or two. The prison can become slightly less claustrophobic, but it is there to stay. All this, however, is bad but hardly unexpected news. The limits of our computers are a direct consequence of the mathematical constraints inherent to classic computation, and these were already known before any PC was built. And yet, Harel stresses that even many members of the computing profession are unaware of these crucial issues. The problem lies in those philosophical arguments that he seems to dismiss as "quasi-scientific". Public resources could be spent more efficiently and scientific efforts could be addressed more accurately if there were a more sound conceptual approach to what computers can really do and how they affect our environment. Instead, the implicit tendency is often to reduce the doable to the "computable". Harel acknowledges the risk but seems to fall into the same trap when he argues, for example, that "the limits of computation are the limits of knowledge". He reduces knowledge to what is deducible from facts, and the latter to what can be computed from facts algorithmically, but the reduction is flawed because it relies on a fundamental ambiguity. "Deducible from facts" can only mean "generally based on facts" in connection with the concept of knowledge, but it must mean "correctly inferred from premises according to mathematical logic" in connection with computation. Human and artificial agents do not share the same constraints, not least because we know this but they cannot. Harel is then right in arguing that showing the limits of our computers means discouraging futile efforts and encouraging the development of new paradigms. And his book is an

excellent antidote. But this is not enough. To make sure that the computational prison does not become our own we also need to develop a better philosophy of computing and ICT, which may provide a foundational investigation into the nature of information and its dynamics (including computation), and guide us in understanding and solving the conceptual as well as normative issues involved in the constitution of our information environments.