Part III

The Read/Write Register Communication Abstraction

This part of the book is devoted to the implementation of read/write registers on top of asynchronous message-passing systems prone to failures. Let us remember that the read/write register is the most basic object in Informatics. It is even the only object of a Turing machine; hence, it is the object sequential computing rests on. This part of the book is composed of five chapters:

- Chapter 5 defines a read/write register in the context of concurrency. It presents three semantics for such an object: regular register, atomic register, and sequentially consistent register. The chapter also shows that $t < n/2$ is a necessary requirement to build a read/write register in the presence of asynchrony and process crashes.
- Chapter 6 is on the implementation of atomic and sequentially consistent read/write registers in $CAMP_{n,t}[t < n/2]$. Multi-Writer/Multi-Reader (MWMR) atomic registers are built incrementally from regular registers. Two approaches for building MWMR sequentially consistent registers are presented.
- Chapter 7 shows how the $t < n/2$ requirement can be circumvented by the use of failure detectors. (Failure detectors have been introduced in Section 3.3. They are "oracles" increasing the computability power of the underlying system by providing information on failures.)
- Chapter 8 presents a specific communication abstraction (called SCD-broadcast), which captures exactly what is needed to implement atomic or sequentially consistent read/write registers. It also shows how this communication abstraction can be implemented in $CAMP_{n,t}[t < n/2]$.
- Chapter 9 presents an implementation of atomic read/write registers in the presence of Byzantine processes. It also shows that such implementations are possible only if $t < n/3$.