Part V
Detection of Properties on Distributed Executions

The two previous parts of the book were on the enrichment of the system to provide processes with high-level operations. Part III was on the definition and the implementation of operations suited to the consistent use of shared resources, while Part IV introduced communication abstractions with specific ordering properties. In both cases, the aim is to allow application programmers to concentrate on their problems and not on the way some operations have to be implemented.

This part of the book is devoted to the observation of distributed computations. Solving an observation problem consists in superimposing a distributed algorithm on a computation, which records appropriate information on this computation in order to be able to detect if it satisfies some property. The specificity of the information is, of course, related to the property one is interested in detecting.

Two detection problems are investigated in this part of the book: the detection of the termination of a distributed execution (Chap. 14), and the detection of deadlocks (Chap. 15). Both properties “the computation has terminated” and “there is deadlock” are stable properties, i.e., once satisfied they remain satisfied in any future state of the computation.

Remark Other property detection problems concern the detection of unstable properties such as the conjunction of local predicates or the detection of properties on execution flows. Their detection is a more advanced topic not covered in this book. The interested reader is invited to consult the following (non-exhaustive) list of references [73, 89, 98, 137, 139, 153, 154, 192, 193, 378].