Chapter XIII
Monadic Second-Order Theories

by Y. Gurevich

In the present chapter we will make a case for the monadic second-order logic (that is to say, for the extension of first-order logic allowing quantification over monadic predicates) as a good source of theories that are both expressive and manageable. We will illustrate two powerful decidability techniques here—the one makes use of automata and games while the other uses generalized products *à la* Feferman-Vaught. The latter is, of course, particularly relevant, since monadic logic definitely appears to be the proper framework for examining generalized products.

Undecidability proofs must be thought out anew in this area; for, whereas true first-order arithmetic is *reducible* to the monadic theory of the real line R, it is nevertheless not *interpretable* in the monadic theory of R. Thus, the examination of a quite unusual undecidability method is another subject that will be explained in this chapter. In the last section we will briefly review the history of the methods thus far developed and give a description of some further results.

1. Monadic Quantification

Monadic (second-order) logic is the extension of the first-order logic that allows quantification over monadic (unary) predicates. Thus, although binary, ternary, and other predicates, as well as functions, may appear in monadic (second-order) languages, they may nevertheless not be quantified over.

1.1. Formal Languages for Mathematical Theories

We are interested less in monadic (second-order) logic itself than in the applications of this logic to mathematical theories. We are interested in the monadic formalization of the language of a mathematical theory and in monadic theories of corresponding mathematical objects. Before we explore this line of thought in more detail, let us argue that formalizing a mathematical language—not necessarily in monadic logic, but rather in first-order logic or in any other formal logic for that matter—can be useful.