Chapter VII
Recursion in a Type-3 Functional

We first hasten to assure the reader that this chapter is not the second in an infinite sequence. Although there are several important differences between the theories of recursion relative to functionals of types 2 and 3, most of the theory of recursion relative to functionals of types greater than 3 can be obtained from type-3 theory with essentially only notational changes. This is discussed in § 4.

In § 1 we consider the basic definitions and facts about recursion in a type-3 functional and examples which illustrate the differences between types 2 and 3. For example, although E and oJ are each recursive in the other, the same is not true of their type-3 analogues $\mathbb{E}$ and $\mathbf{s}\mathbb{J}$. In § 2 we see that although the basic structure of the class of relations semi-recursive in a type-3 functional is superficially similar to the corresponding structure of type-2, the differences begin to be more important. Finally in § 3 we see that with respect to hierarchies the situation for recursion in a type-3 functional is very different from that for type-2.

1. Basic Properties

We shall consider in detail only the notion of recursion relative to a single fixed total type-3 functional $\mathbb{I}$: ${}^{({}^{\omega}\omega)}\omega \to \omega$. From this may easily be derived, by the usual sorts of coding, notions of recursion relative to several type-3 functionals and, as in § VI.7, the notion of a recursive type-4 functional. Although we are still primarily interested in the properties of relations and functionals over ${}^{k,l}\omega$, we shall also need the notion of relative recursiveness among type-3 functionals. For this reason we state the basic definitions in terms of functionals and relations over ${}^{k,l,l'}\omega$.

Consider first the intuitive notion of a functional $\mathbb{F}$ being calculable relative to $\mathbb{I}$. We must stretch our imagination one step further to conceive of an idealized computer prepared to receive inputs of the form $(\mathbf{m}, \boldsymbol{\alpha}, \mathbf{I})$ and connected to a memory device M which contains the graph of $\mathbb{I}$. The inputs are considered to be stored before the beginning of the computation in infinite memory devices,