Chapter 2

General Theory: Subcomputations

This chapter adds a new notion to the general theory, viz. the notion of *subcomputation.* We develop the elementary theory including a general version of the first recursion theorem, and ending up with a representation theorem which is "faithful" in the sense that it preserves the full structure of subcomputations.

2.1 Subcomputations

In Chapter 1 we took as our basic relation

$$\{a\}(\sigma) \simeq z,$$

asserting that the computing device a acting on the input sequence σ gives z as output. We wrote down for the set Θ of all computation tuples (a, σ, z) a set of axioms and were able to derive within this framework a number of results of elementary recursion theory, leading up to a *simple representation theorem* for any such Θ.

However, many arguments from the more advanced parts of recursion theory seem to require an analysis not only of the computation tuple, but of the whole structure of "subcomputations" of a given computation tuple. In fact, such an analysis was involved in the proof of the first recursion Theorem 1.7.9 *via* the representation Theorem 1.6.3 (see Definition 1.5.9).

In his paper *Axioms for computation theories—first draft* [113] Moschovakis emphasized the fact that whatever computations may be, they have assigned a well-defined *length*, which is always an ordinal, finite or infinite. Thus he proposed to add as a further primitive a map from the set Θ of computation tuples to the ordinals, denoting by $|a, \sigma, z|_\Theta$ the ordinal associated with the tuple $(a, \sigma, z) \in \Theta$.

We shall, in addition, abstract another but related aspect of the notion of computation and add as a further primitive a relation between computation tuples

$$(a', \sigma', z') < (a, \sigma, z),$$

which is intended to express that (a', σ', z') is a "subcomputation" of (a, σ, z), i.e. the computation (a, σ, z) depends upon the previous computation (a', σ', z').