

Chapter XIII

Decomposition Theorems and Weight

In this chapter we will show every type in a superstable theory can be decomposed in terms of a finite number of regular types. In fact, under a suitable operation we impose a structure on the stationary types which reflects the multiplicative structure of the natural numbers. The weight one types behave as primes in this representation. We first obtain a precise structure theory for finitely generated extensions of an \mathbf{S} -model. We connect these structural results with the notion of weight. While we would like to develop such a decomposition theorem for models in an arbitrary acceptable class K , we can not do so uniformly. Rather, we first obtain the result for \mathbf{S} -models in Section 1 then define weight in Section 2. We conclude Section 2 by invoking the notion of weight to prove Lachlan's theorem that a countable superstable theory has either 1 or infinitely many countable models. In Section 3 we show that in an ω -stable theory there are 'enough' \mathbf{AT} -strongly regular types. With this tool we obtain an extension of the decomposition theorem to all models of an ω -stable theory in Section 4.

Except for a few results at the beginning of Section 2, we assume in this chapter that T is superstable.

1. The Decomposition Theorem For \mathbf{S} -Models

In this section we restrict ourselves to the class of \mathbf{S} -models. We show that each finitely generated \mathbf{S} -model of a superstable theory has a well defined dimension. We will use this information to decompose all types in a superstable theory as a product of regular types. The results of this section provide one characterization of weight. In Section 4, we consider extending the results of this section to other classes K .

1.1 Definition. Let M be an \mathbf{S} -model, $A \subseteq M$, and let $R(M, A)$ be the collection of points in M which realize stationary regular types over A . Then $\dim(R(M, A))$ is the cardinality of a maximal independent subset of $R(M, A)$.