questions. Here there seems to be no way to proceed without Church's Thesis. Second, we sometimes use Church's Thesis to prove a function is recursive by observing that it is computable and using Church's Thesis to conclude that it is recursive. This type of use is non–essential; we could always use the methods we have developed to prove that the function is recursive. One of the best ways to convince oneself of Church's Thesis is to examine many such examples and see that in every case the function turns out to be recursive.

10. Word Problems

The initial aim of recursion theory was to show that certain problems of the form "Find an algorithm by which ..." were unsolvable. We shall give a few examples of such problems.

Let us first see how to obtain a non–recursive real F. By 8.1, it is enough to make F different from each $\{e\}$. We shall do this by making it different from $\{e\}$ at the argument e. (This idea, known as the diagonal argument, was used first by Cantor to prove that the set of real numbers is uncountable.) In more detail, we define

$$\begin{aligned} F(e) &\simeq \{e\}(e) + 1 && \text{if } \{e\}(e) \text{ is defined,} \\ &\simeq 0 && \text{otherwise.} \end{aligned}$$

It follows from this construction that the set P defined by

$$P(e) \leftrightarrow \{e\}(e) \text{ is defined}$$

is not recursive; for otherwise, the definition of F would be a definition by cases using only recursive symbols, and hence F would be recursive. Thus, using Church's Thesis, we have our first unsolvable problem: find an algorithm for deciding if $\{e\}(e)$ is defined.

Consider the following problem, called the halting problem: Find an algorithm by which, given a program P and an x, we can decide if the computation of P from x halts. Let P be a program which computes the re-