Chapter 15
The Theme of This Book: Delaying Tactics

There is no sense of urgency in the construction of c.e. sets.[1] We do not care how much time a stage takes, as long as it is finite. Nor do we care how many stages we must wait for an event to happen, as long as that, too, is a finite number.

When all finite amounts of time are regarded as the same, delaying tactics do not hurt, but they might help. Indeed, many of the key algorithmic ideas in this book are delaying tactics; as examples:

1. When permitting is used, a witness must wait for permission to enter the set under construction.
2. A node σ in a priority tree is idle except during σ-stages.
3. During each stage s such that $TP_s <_L \sigma$, a node σ in a priority tree gets re-initialized or canceled in some way, thereby (typically) wiping out any progress that it has already made toward meeting its associated requirements. Fortunately, if $\sigma \in TP$ then there are infinitely many σ-stages, but only finitely many such cancellations.
4. In Chaps. 11 and 12, we use the node-specific computation $\Phi_\sigma^B(k)[s]$, rather than $\Phi_e^B(k)[s]$, to define the length of agreement. In other words, we pay attention to the computation $\Phi_e^B(k)[s]$ only if it is σ-believable. This has the side effect of slowing down (but not stopping) the growth of the length of agreement, if $\sigma \in TP$.
5. In Chap. 14, the star witness concept has the (harmless) side effect of slowing down but not stopping the growth of a witness list.
   Likewise in Chap. 14, both unrealization and insertion take precedence over realization. This, too, might slow down (but not stop) the growth of a witness list.

   Both of these delaying tactics are discussed in Note 7 of Sect. 3.3 of Chap. 14.

[1] Hence there is no mention of fast data structures in this book.

K. J. Supowit, *Algorithms for Constructing Computably Enumerable Sets*,
Computer Science Foundations and Applied Logic,
https://doi.org/10.1007/978-3-031-26904-2_15