Figure 11.11 A hierarchical planning algorithm that uses angelic semantics to identify and commit to high-level plans that work while avoiding high-level plans that don’t. The predicate MAKING-PROGRESS checks to make sure that we aren’t stuck in an infinite regression of refinements. At top level, call ANGELIC-SEARCH with [Act] as the initialPlan.

The ability to commit to or reject high-level plans can give ANGELIC-SEARCH a significant computational advantage over HIERARCHICAL-SEARCH, which in turn may have a large advantage over plain old BREADTH-FIRST-SEARCH. Consider, for example, cleaning up a large vacuum world consisting of an arrangement of rooms connected by narrow corridors, where each room is a $w \times h$ rectangle of squares. It makes sense to have an HLA for Navigate (as shown in Figure 11.7) and one for CleanWholeRoom. (Cleaning the room could be implemented with the repeated application of another HLA to clean each row.) Since there are five primitive actions, the cost for BREADTH-FIRST-SEARCH grows as $5^d$, where $d$ is the length of the shortest solution (roughly twice the total number of squares); the algorithm cannot manage even two $3 \times 3$ rooms. HIERARCHICAL-SEARCH is more efficient, but still suffers from exponential growth because it tries all ways of cleaning that are consistent with the hierarchy. ANGELIC-SEARCH scales approximately linearly in the number of squares—it commits to a good high-level sequence of room-cleaning and navigation steps and prunes away the other options.