```
function ANGELIC-SEARCH(problem, hierarchy, initialPlan) returns a solution or fail

  frontier ← a FIFO queue with initialPlan as the only element
  while true do
      if IS-EMPTY?(frontier) then return fail
      plan ← POP(frontier)          // chooses the shallowest node in frontier
      if REACH+(problem.INITIAL, plan) intersects problem.GOAL then
          if plan is primitive then return plan          // REACH+ is exact for primitive plans
          guaranteed ← REACH−(problem.INITIAL, plan) ∩ problem.GOAL
          if guaranteed≠{ } and MAKING-PROGRESS(plan, initialPlan) then
              finalState ← any element of guaranteed
              return DECOMPOSE(hierarchy, problem.INITIAL, plan, finalState)
          hla ← some HLA in plan
          prefix,suffix ← the action subsequences before and after hla in plan
          outcome ← RESULT(problem.INITIAL, prefix)
          for each sequence in REFINEMENTS(hla, outcome, hierarchy) do
              add APPEND(prefix, sequence, suffix) to frontier

function DECOMPOSE(hierarchy, s0, plan, sf) returns a solution

  solution ← an empty plan
  while plan is not empty do
     action ← REMOVE-LAST(plan)
     si ← a state in REACH−(s0, plan) such that sf ∈ REACH−(si, action)
     problem ← a problem with INITIAL = si and GOAL = sf
     solution ← APPEND(ANGELIC-SEARCH(problem, hierarchy, action), solution)
     sf ← si
  return solution
```