```
function CPOP(initial, goals, operators) returns plan

  plan ← MAKE-PLAN(initial, goals)
  loop do

     Termination:
        if there are no unsatisfied preconditions
           and the contexts of the finish steps are exhaustive
            then return plan

     Alternative context generation:
        if the plans for existing finish steps are complete and have contexts C_1 ... C_n then
            add a new finish step with a context ¬ (C_1 ∨ ... ∨ C_n)
            this becomes the  current context

     Subgoal selection and addition:
        find a plan step S_need with an open precondition c

     Action selection:
        choose a step S_add from operators or STEPS(plan) that adds c or
                knowledge of c and has a context compatible with the current context
        if there is no such step
            then fail
        add S_add --c--> S_need to LINKS(plan)
        add S_add < S_need to ORDERINGS(plan)
        if S_add is a newly added step then
            add S_add to STEPS(plan)
            add Start < S_add < Finish to ORDERINGS(plan)

     Threat resolution:
        for each step S_threat that potentially threatens any causal link S_i --c--> S_j
                 with a compatible context do
            choose one of
                 Promotion: Add S_threat < S_i to ORDERINGS(plan)
                 Demotion: Add S_j < S_threat to ORDERINGS(plan)
                 Conditioning:
                     find a conditional step S_cond possibly before both S_threat and S_j, where
                          1. the context of S_cond is compatible with the contexts of S_threat and S_j;
                          2. the step has outcomes consistent with S_threat and S_j, respectively
                     add conditioning links for the outcomes from S_cond to S_threat and S_j
                     augment and propagate the contexts of S_threat and S_j

            if no choice is consistent
                     then fail
        end
  end
```