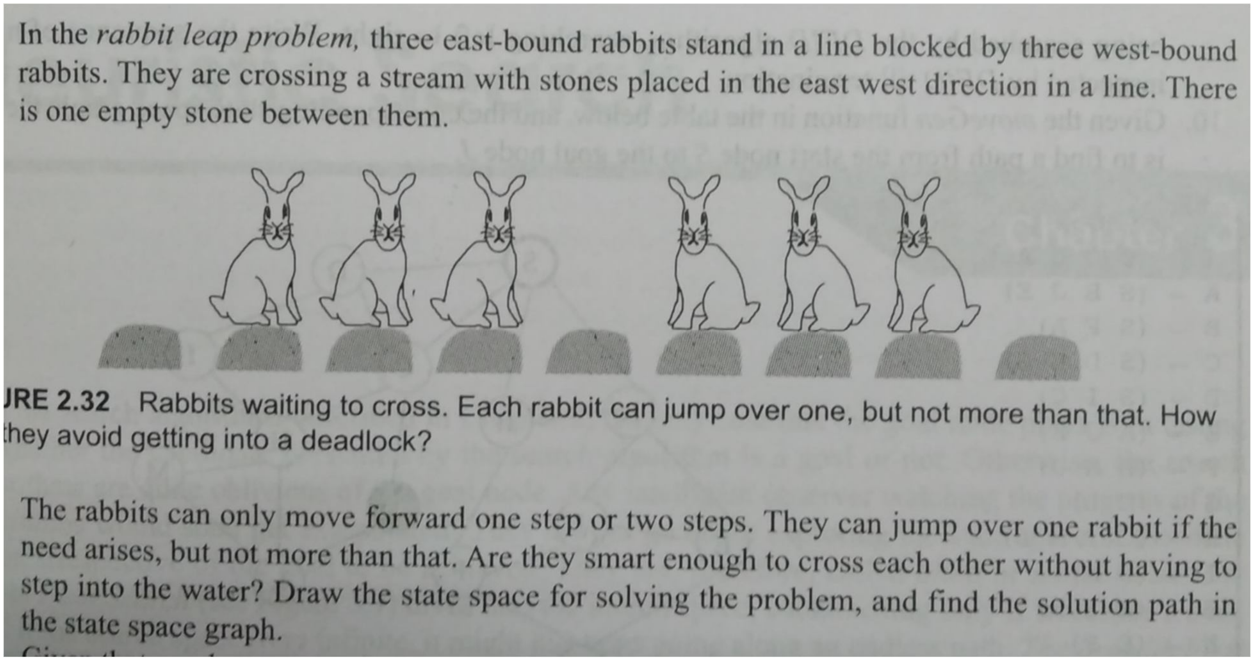
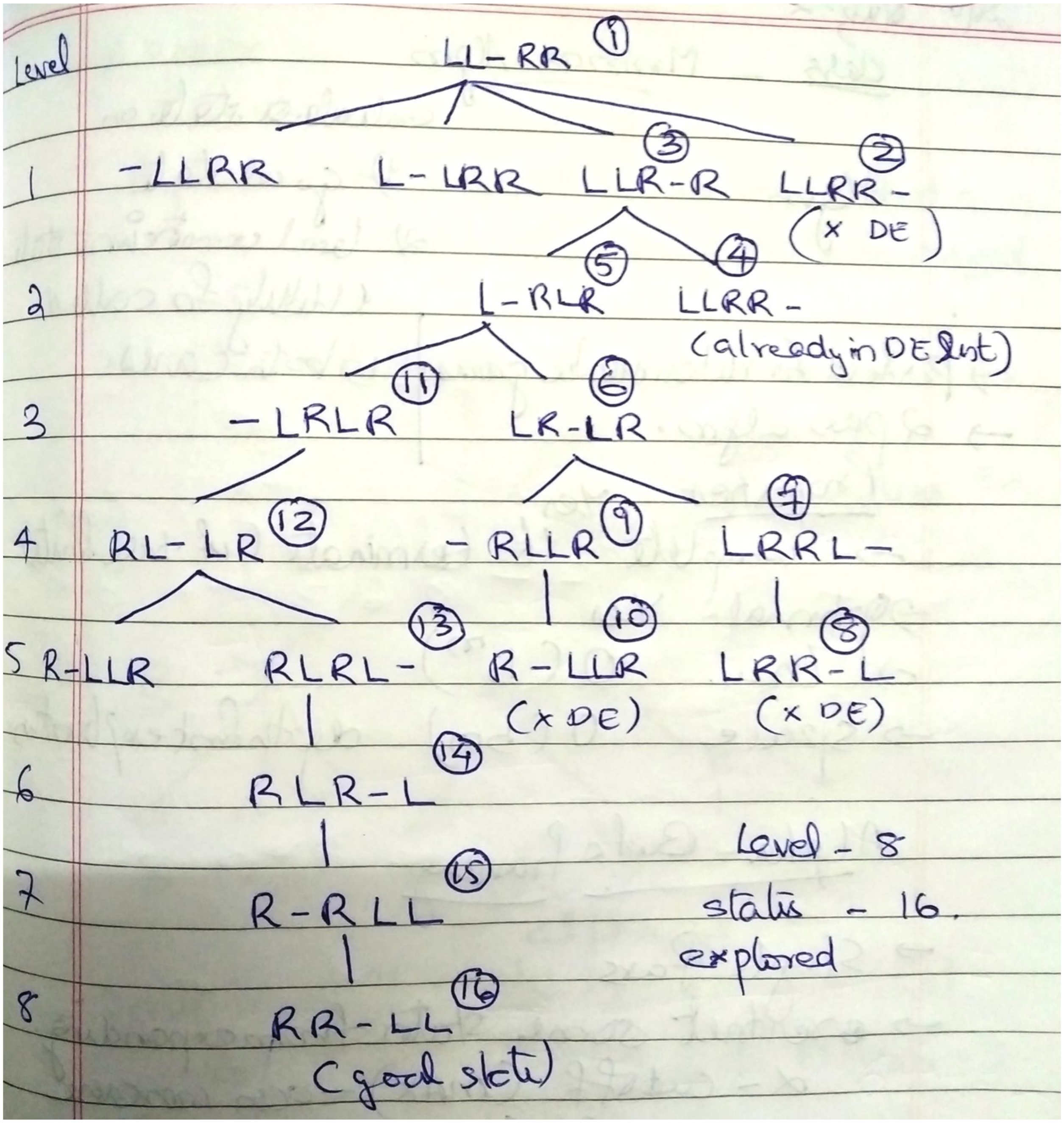


In the *rabbit leap problem,* three east-bound rabbits stand in a line blocked by three west-bound rabbits. They are crossing a stream with stones placed in the east west direction in a line. There is one empty stone between them.

URE 2.32 Rabbits waiting to cross. Each rabbit can jump over one, but not more than that. How they avoid getting into a deadlock?

The rabbits can only move forward one step or two steps. They can jump over one rabbit if the need arises, but not more than that. Are they smart enough to cross each other without having to step into the water? Draw the state space for solving the problem, and find the solution path in the state space graph.



Level - 8
states - 16.
explored

Implementing Graph Search: Function backtrack algorithm

```
function backtrack;

begin
  SL := [Start];  NSL := [Start];  DE := [ ];  CS := Start;          % initialize:
  while NSL ≠ [ ] do                              % while there are states to be tried
    begin
      if CS = goal (or meets goal description)
        then return SL;                  % on success, return list of states in path.
      if CS has no children (excluding nodes already on DE, SL, and NSL)
        then begin
          while SL is not empty and CS = the first element of SL do
            begin
              add CS to DE;                             % record state as dead end
              remove first element from SL;                          %backtrack
              remove first element from NSL;
              CS := first element of NSL;
            end
          add CS to SL;
        end
        else begin
          place children of CS (except nodes already on DE, SL, or NSL) on NSL;
          CS := first element of NSL;
          add CS to SL
        end
    end;
    return FAIL;
end.
```