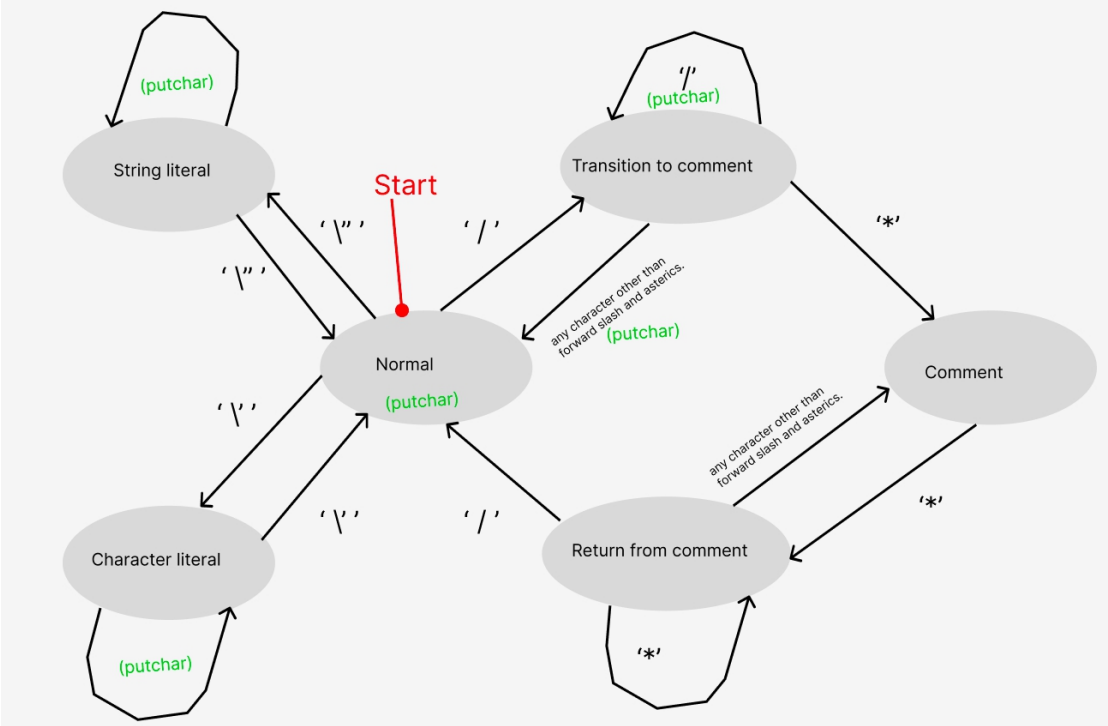

This is the DFA design.

There are 6 different states according to my DFA design for decomment program.

Normal is the beginning of the state while reading the c code. It puts every character entered to the out put c file except.

1. When ‘/’ character is encountered since it is a possible way to enter into comment state the state is changed to transition to comment state. Then:

a. If asterics is incountered in transition to comment state the state changes to comment state. And stops putting characters as it’s supposed to do.

b. If / is encountered it remains in transition to comment state putting the previous / character encountered.

c. If any character other than asterics and forward slash characters is entered, forward slash and the entered character are putted and the state is changed to normal state.

2. In comment state characters are not putted to the output file. When * character is encountered the state changes to return from comment state. Then

a. If ‘/’ is encountered the state is changes to Normal state and will begin putting characters based on specified in 1.

b. if ‘*’ is encountered the state remains in Return from comment state since if ‘/’ is encountered in the next iteration the state could change to Normal state.

c. If any character other than the forward slash and asterics is encountered the state returns to comment state.

3. When (‘\” ’) is encountered the state changes to string literal state. Here it puts every character encountered in the state. The state checks for (‘\” ’) to be encountered again to return to Normal state. Transition to string literal state is only possible if the state is in Normal state.

4. When (‘ \’ ’) is encountered the state changes to Character literal state. Here it puts every character encountered in the state. The state checks for ‘ \’ ’) to be encountered again to return to Normal state. And likewise, transition to character literal state is only possible if the state is in Normal state.