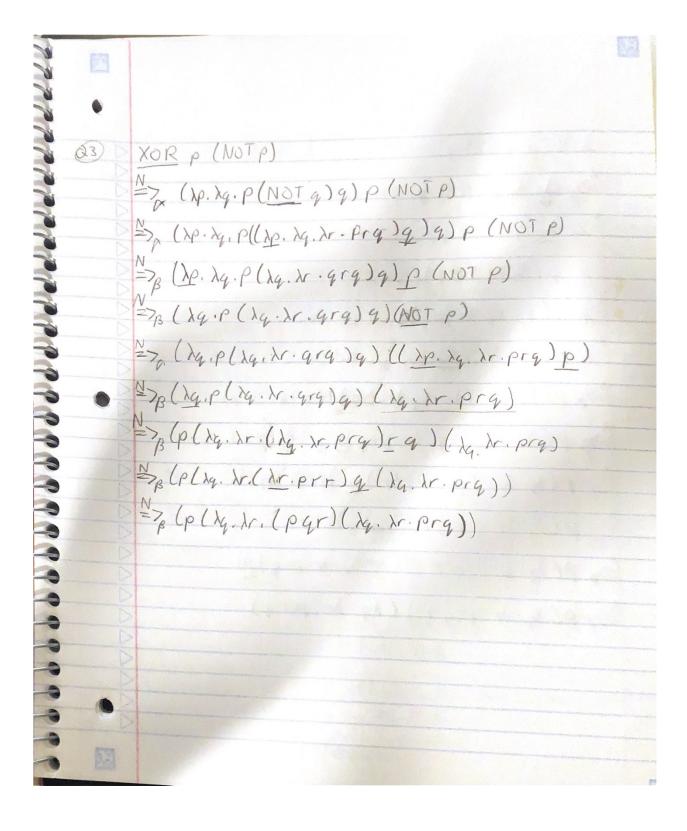
1) The two expression operands are always evaluated when using the logical XOR operator, therefore, it cannot be short-circuited. unless you consider lazy evaluation as a proper way to short circuit XOR, it would look something like this... if (not A) return B; if (not B) return A; return 0; would be short-circuiting because returning B doesn't require evaluating B. (lazy eval) 2) function recursiveWhile(input) { if (condition according to input) { // execute "while loop" // call recursiveWhile with the same/updated condition until condition is false recursiveWhile(input); } else { // exit recursion to main } } // start the recursion with an initial input recursiveWhile(initialCondition);

3)



XOR P (NOTP) => (Ap. 24. P (NOT 4) 9) P(NOT P) = 7 p /g . p (NOT g) g (NOT p) 2)B P(NOT(NOTP)(NOTP) => p(l/p. hq. hr. prq) (NOT p) (NOT P) => p (hg. hr. (NOTP) rg) (NOT p) = Top p(hg. hr. (hp. hg. hr. pra)prg) (NOT p) =>BP(2g. Ar. (Ag. Ar. Pry)rg)(Notp) Explandr. (dr. priri) g) (NOTP) =>BP(kg. hr. pgr)(NOTp) => p(kg. hr. pgr)((hp. kg. hr. prg)p) =>pp(dq. dr.pgr)(dg.dr.prg) 阿賈

