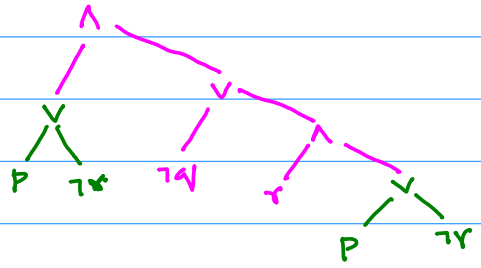
932, 933, 934, 935

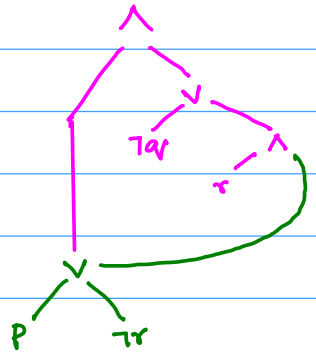
Normal forms

Negation Normal Form : $(p \vee \neg r) \wedge (\neg q \vee (r \wedge (p \vee \neg r)))$

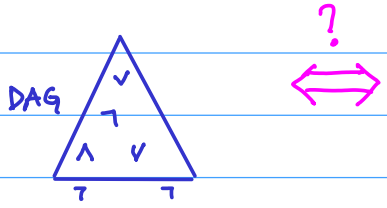
Tree representation:



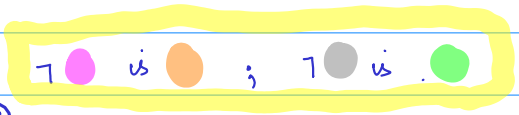
DAG representation:

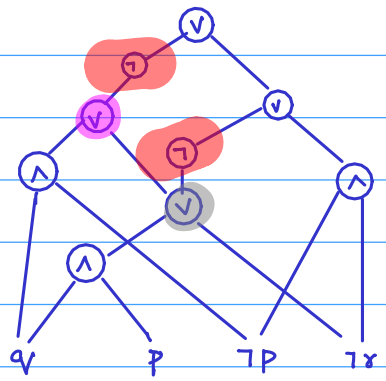


Given a DAG representing a prop. logic formula with only $\wedge, \vee, \neg$ nodes, can we efficiently get a DAG representing a semantically equivalent NNF formula?

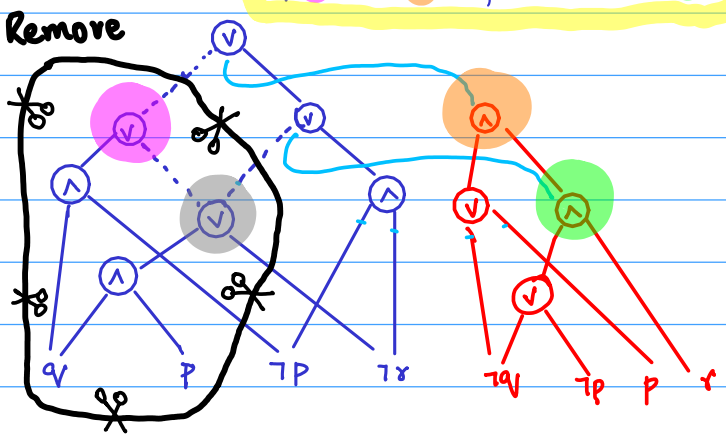


?





?



Literal : variable or its complement $p, \neg p, q, \neg r$

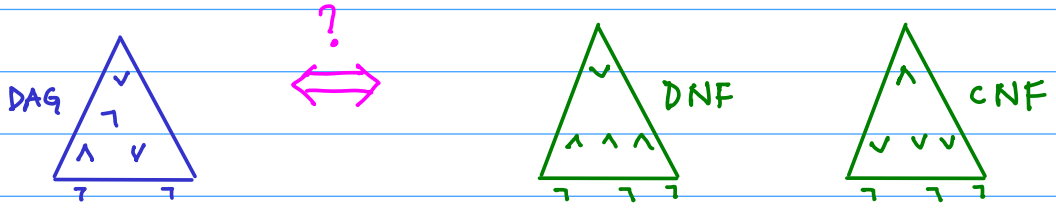
Clause : Disjunction of literals $(p \vee q \vee \neg r)$, ~~$(q \vee \neg q \vee r)$~~

Cube : Conjunction of literals $(p \wedge q \wedge \neg r)$, ~~$(q \wedge \neg q \wedge r)$~~

Conjunctive Normal Form (CNF) : Conjunction of clauses
(Product of Sums) $(p \vee q \vee \neg r) \wedge (q \vee r \vee \neg s \vee t)$

Disjunctive Normal Form (DNF) : Disjunction of cubes
(Sum of Products) $(\neg p \wedge \neg q \wedge r) \vee (\neg q \wedge \neg r \wedge s \wedge \neg t)$

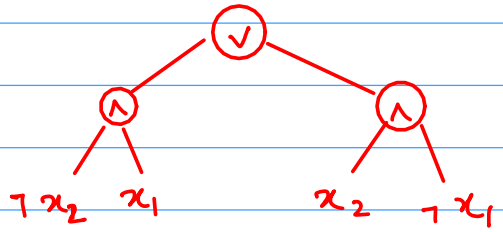
Given a DAG representing a prop. logic formula with only $\wedge, \vee, \neg$ nodes, can we efficiently get a DAG representing a semantically equivalent ~~NNF~~ CNF/DNF formula?



Validity checking of { CNF, DNF }

Satisfiability checking of { CNF, DNF }

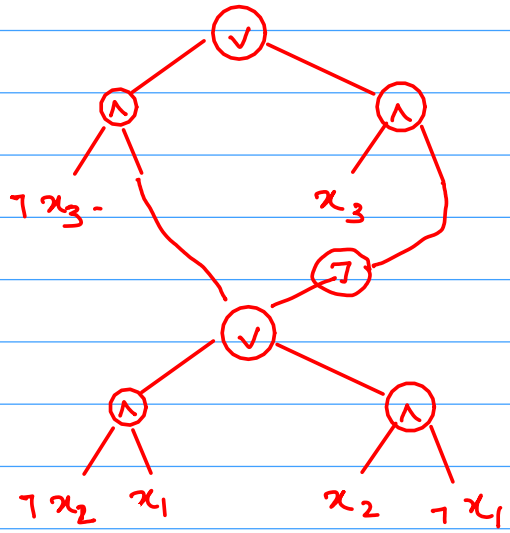
$x_1 \oplus x_2 \Leftrightarrow (x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2)$



$(x_1 \oplus x_2) \oplus x_3$

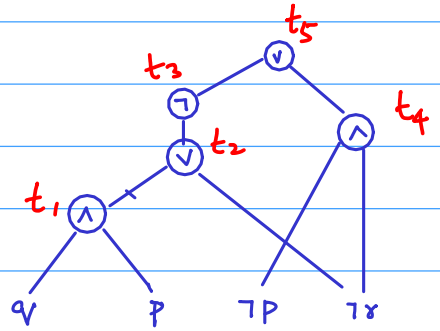
$\varphi \quad \oplus x_3$

$4n$ nodes



$\varphi(p, q, r)$

$\varphi'(p, q, r, t_1 \ldots t_5)$ Equisatisfiable. CNF



$(t_1 \leftrightarrow p \wedge q) \wedge$

$(t_2 \leftrightarrow t_1 \vee \neg r) \wedge$

$(t_3 \leftrightarrow \neg t_2) \wedge$

$(t_4 \leftrightarrow \neg p \wedge \neg r) \wedge$

$(t_5 \leftrightarrow t_3 \vee t_4) \wedge t_5$

$(t_1 \rightarrow (p \wedge q)) \wedge ((p \wedge q) \rightarrow t_1)$

$\Leftrightarrow (\neg t_1 \vee (p \wedge q)) \wedge (\neg(p \wedge q) \vee t_1) \Leftrightarrow (\neg t_1 \vee p) \wedge (\neg t_1 \vee q) \wedge (\neg p \vee \neg q \vee t_1)$